

BIROn - Birkbeck Institutional Research Online

Gollini, Isabella and Lu, B. and Charlton, M. and Brunsdon, C. and Harris, P. (2015) GWmodel: an R package for exploring spatial heterogeneity using geographically weighted models. Journal of Statistical Software 63 (17), ISSN 1548-7660.

Downloaded from: <https://eprints.bbk.ac.uk/id/eprint/13189/>

Usage Guidelines:

Please refer to usage guidelines at <https://eprints.bbk.ac.uk/policies.html>
contact lib-eprints@bbk.ac.uk.

or alternatively



GWmodel: An R Package for Exploring Spatial Heterogeneity Using Geographically Weighted Models

Isabella Gollini
University of Bristol

Binbin Lu
Wuhan University

Martin Charlton
NUI Maynooth

Christopher Brunsdon
NUI Maynooth

Paul Harris
Rothamsted Research

Abstract

Spatial statistics is a growing discipline providing important analytical techniques in a wide range of disciplines in the natural and social sciences. In the R package **GWmodel**, we present techniques from a particular branch of spatial statistics, termed geographically weighted (GW) models. GW models suit situations when data are not described well by some global model, but where there are spatial regions where a suitably localized calibration provides a better description. The approach uses a moving window weighting technique, where localized models are found at target locations. Outputs are mapped to provide a useful exploratory tool into the nature of the data spatial heterogeneity. Currently, **GWmodel** includes functions for: GW summary statistics, GW principal components analysis, GW regression, and GW discriminant analysis; some of which are provided in basic and robust forms.

Keywords: geographically weighted regression, geographically weighted principal components analysis, spatial prediction, robust, R package.

1. Introduction

Spatial statistics provides important analytical techniques for a wide range of disciplines in the natural and social sciences, where (often large) spatial data sets are routinely collected. Here we present techniques from a particular branch of non-stationary spatial statistics, termed geographically weighted (GW) models. GW models suit situations when spatial data are not

described well by some universal or global model, but where there are spatial regions where a suitably localized model calibration provides a better description. The approach uses a moving window weighting technique, where localized models are found at target locations. Here, for an individual model at some target location, we weight all neighboring observations according to some distance-decay kernel function and then locally apply the model to this weighted data. The size of the window over which this localized model might apply is controlled by the bandwidth. Small bandwidths lead to more rapid spatial variation in the results while large bandwidths yield results increasingly close to the universal model solution. When there exists some objective function (e.g., the model can predict), a bandwidth can be found optimally, using cross-validation and related approaches.

The GW modelling paradigm has evolved to encompass many techniques; techniques that are applicable when a certain heterogeneity or non-stationarity is suspected in the study's spatial process. Commonly, outputs or parameters of the GW model are mapped to provide a useful exploratory tool, which can often precede (and direct) a more traditional or sophisticated statistical analysis. Subsequent analyses can be non-spatial or spatial, where the latter can incorporate stationary or non-stationary decisions. Notable GW models include: GW summary statistics (Brunsdon, Fotheringham, and Charlton 2002); GW principal components analysis (GW PCA, Fotheringham, Brunsdon, and Charlton 2002; Lloyd 2010a; Harris, Brunsdon, and Charlton 2011a); GW regression (Brunsdon, Fotheringham, and Charlton 1996, 1998, 1999; Leung, Mei, and Zhang 2000; Wheeler 2007); GW generalized linear models (Fotheringham *et al.* 2002; Nakaya, Fotheringham, Brunsdon, and Charlton 2005); GW discriminant analysis (Brunsdon, Fotheringham, and Charlton 2007); GW variograms (Harris, Charlton, and Fotheringham 2010a); GW regression kriging hybrids (Harris and Juggins 2011) and GW visualization techniques (Dykes and Brunsdon 2007).

Many of these GW models are included in the R (R Core Team 2014) package **GWmodel** that we describe in this paper. Those that are not, will be incorporated at a later date. For the GW models that are included, there is a clear emphasis on data exploration. Notably, **GWmodel** provides functions to conduct: (i) a GW PCA; (ii) GW regression with a local ridge compensation (for addressing local collinearity); (iii) mixed GW regression; (iv) heteroskedastic GW regression; (v) a GW discriminant analysis; (vi) robust and outlier-resistant GW modelling; (vii) Monte Carlo significance tests for non-stationarity; and (viii) GW modelling with a wide selection of distance metric and kernel weighting options. These functions extend and enhance functions for: (a) GW summary statistics; (b) basic GW regression; and (c) GW generalized linear models – GW models that are also found in the **spgwr** R package (Bivand, Yu, Nakaya, and Garcia-Lopez 2013). In this respect, **GWmodel** provides a more extensive set of GW modelling tools, within a single coherent framework (**GWmodel** similarly extends or complements the **gwrr** R package (Wheeler 2013b) with respect to GW regression and local collinearity issues). **GWmodel** also provides an advanced alternative to various executable software packages that have a focus on GW regression – such as **GWR 3** (Charlton, Fotheringham, and Brunsdon 2003); the **ArcGIS** GW regression tool in the Spatial Statistics Toolbox (ESRI 2013); **SAM** for GW regression applications in macroecology (Rangel, Diniz-Filho, and Bini 2010); and **SpaceStat** for GW regression applications in health (BioMedware 2011).

Noting that it is not feasible to describe in detail all of the available functions in **GWmodel**, our paper has a robust theme and is structured as follows. Section 2 describes the example data sets that are available in **GWmodel**. Section 3 describes the various distance metric and

kernel weighting options. Section 4 describes modelling with basic and robust GW summary statistics. Section 5 describes modelling with basic and robust GW PCA. Section 6 describes modelling with basic and robust GW regression. Section 7 describes ways to address local collinearity issues when modelling with GW regression. Section 8 describes how to use GW regression as a spatial predictor. Section 9 relates the functions of **GWmodel** to those found in the **spgwr**, **gwrr** and **McSpatial** (McMillen 2013) R packages. Section 10 concludes this work and indicates future work.

2. Data sets

The **GWmodel** package comes with five example data sets, these are: (i) **Georgia**, (ii) **LondonHP**, (iii) **USelect**, (iv) **DubVoter**, and (v) **EWHP**. The **Georgia** data consists of selected 1990 US census variables (with $n = 159$) for counties in the US state of Georgia; and is fully described in Fotheringham *et al.* (2002). This data has been routinely used in a GW regression context for linking educational attainment with various contextual social variables (see also Griffith 2008). The data set is also available in the **GWR** 3 executable software package (Charlton *et al.* 2003) and the **spgwr** R package.

The **LondonHP** data is a house price data set for London, England. This data set (with $n = 372$) is sampled from a 2001 house price data set, provided by the Nationwide Building Society of the UK and is combined with various hedonic contextual variables (Fotheringham *et al.* 2002). The hedonic data reflect structural characteristics of the property, property construction time, property type and local household income conditions. Studies in house price markets with respect to modelling hedonic relationships has been a common application of GW regression (e.g., Kestens, Thériault, and Rosiers 2006; Bitter, Mulligan, and Dall’Erba 2007; Páez, Long, and Farber 2008).

The **USelect** data consists of the results of the 2004 US presidential election at the county level, together with five census variables (with $n = 3111$). The data is a subset of that provided in (Robinson 2013). **USelect** is similar to that used for the visualization of GW discriminant analysis outputs in Foley and Demsar (2013); the only difference is that we specify the categorical, election results variable with three classes (instead of two): (a) Bush winner, (b) Kerry winner and (c) Borderline (for marginal winning results).

For this article’s presentation of GW models, we use as case studies, the **DubVoter** and **EWHP** data sets. The **DubVoter** data (with $n = 322$) is the main study data set and is used throughout Sections 4 to 7, where key GW models are presented. This data is composed of nine percentage variables¹, measuring: (1) voter turnout in the Irish 2004 Dáil elections and (2) eight characteristics of social structure (census data); for 322 Electoral Divisions (EDs) of Greater Dublin. Kavanagh, Fotheringham, and Charlton (2006) modelled this data using GW regression; with voter turnout (**GenEl2004**) the dependent variable (i.e., the percentage of the population in each ED who voted in the election). The eight independent variables measure the percentage of the population in each ED, with respect to:

- **DiffAdd**: One year migrants (i.e., moved to a different address one year ago).
- **LARent**: Local authority renters.

¹Observe that none of the **DubVoter** variables constitute a closed system (i.e., the full array of values sum to 100) and as such, we do not need to transform the data prior to a (univariate or multivariate) GW model calibration.

- **SC1**: Social class one (high social class).
- **Unempl**: Unemployed.
- **LowEduc**: Without any formal educational.
- **Age18_24**: Age group 18–24.
- **Age25_44**: Age group 25–44.
- **Age45_64**: Age group 45–64.

Thus the eight independent variables reflect measures of migration, public housing, high social class, unemployment, educational attainment, and three adult age groups.

The EWHP data (with $n = 519$) is a house price data set for England and Wales, this time sampled from 1999, but again provided by the Nationwide Building Society and combined with various hedonic contextual variables. Here for a regression fit, the dependent variable is **PurPrice** (what the house sold for) and the nine independent variables are: **BldIntWr**, **BldPostW**, **Bld60s**, **Bld70s**, **Bld80s**, **TypDetch**, **TypSemiD**, **TypFlat** and **FlrArea**. All independent variables are indicator variables (1 or 0) except for **FlrArea**. Section 8 uses this data when demonstrating GW regression as a spatial predictor; where **PurPrice** is considered as a function of **FlrArea** (house floor area), only.

3. Distance matrix, kernel and bandwidth

A fundamental element in GW modelling is the spatial weighting function (Fotheringham *et al.* 2002) that quantifies (or sets) the spatial relationship or spatial dependency between the observed variables. Here $W(u_i, v_i)$ is a $n \times n$ (with n the number of observations) diagonal matrix denoting the geographical weighting of each observation point for model calibration point i at location (u_i, v_i) . We have a different diagonal matrix for each model calibration point. There are three key elements in building this weighting matrix: (i) the type of distance, (ii) the kernel function and (iii) its bandwidth.

3.1. Selecting the distance function

Distance can be calculated in various ways and does not have to be Euclidean. An important family of distance metrics are Minkowski distances. This family includes the usual Euclidean distance having $p = 2$ and the Manhattan distance when $p = 1$ (where p is the power of the Minkowski distance). Another useful metric is the great circle distance, which finds the shortest distance between two points taking into consideration the natural curvature of the Earth. All such metrics are possible in **GWmodel**.

3.2. Kernel functions and bandwidth

A set of commonly used kernel functions are shown in Table 1 and Figure 1; all of which are available in **GWmodel**. The ‘Global Model’ kernel, that gives a unit weight to each observation, is included in order to show that a global model is a special case of its GW model.

Global Model	$w_{ij} = 1$
Gaussian	$w_{ij} = \exp\left(-\frac{1}{2}\left(\frac{d_{ij}}{b}\right)^2\right)$
Exponential	$w_{ij} = \exp\left(-\frac{ d_{ij} }{b}\right)$
Box-car	$w_{ij} = \begin{cases} 1 & \text{if } d_{ij} < b, \\ 0 & \text{otherwise} \end{cases}$
Bi-square	$w_{ij} = \begin{cases} (1 - (d_{ij}/b)^2)^2 & \text{if } d_{ij} < b, \\ 0 & \text{otherwise} \end{cases}$
Tri-cube	$w_{ij} = \begin{cases} (1 - (d_{ij} /b)^3)^3 & \text{if } d_{ij} < b, \\ 0 & \text{otherwise} \end{cases}$

Table 1: Six kernel functions; w_{ij} is the j -th element of the diagonal of the matrix of geographical weights $W(u_i, v_i)$, and d_{ij} is the distance between observations i and j , and b is the bandwidth.

The Gaussian and exponential kernels are continuous functions of the distance between two observation points (or an observation and calibration point). The weights will be a maximum (equal to 1) for an observation at a GW model calibration point, and will decrease according to a Gaussian or exponential curve as the distance between observation/calibration points increases.

The box-car kernel is a simple discontinuous function that excludes observations that are further than some distance b from the GW model calibration point. This is equivalent to setting their weights to zero at such distances. This kernel allows for efficient computation, since only a subset of the observation points need to be included in fitting the local model at each GW model calibration point. This can be particularly useful when handling large data sets.

The bi-square and tri-cube kernels are similarly discontinuous, giving null weights to observations with a distance greater than b . However unlike a box-car kernel, they provide weights that decrease as the distance between observation/calibration points increase, up until the distance b . Thus these are both distance-decay weighting kernels, as are Gaussian and exponential kernels.

The key controlling parameter in all kernel functions is the bandwidth b . For the discontinuous functions, bandwidths can be specified either as a fixed distance or as a fixed number of local data (i.e., an adaptive distance). For the continuous functions, bandwidths can be specified either as a fixed distance or as a ‘fixed quantity that reflects local sample size’ (i.e., still an ‘adaptive’ distance, but the actual local sample size will be the sample size, as functions are continuous). In practise a fixed bandwidth suits fairly regular sample configurations whilst an adaptive bandwidth suits highly irregular sample configurations. Adaptive bandwidths ensure sufficient (and constant) local information for each local calibration of a given GW model. Bandwidths for GW models can be user-specified or found via some automated (e.g., cross-validation) procedure provided some objective function exists. Specific functions (`bw.gwr`, `bw.gwr.lcr`, `bw.ggwr`, `bw.gwpca`, `bw.gwda`) can be used to find such optimal bandwidths, depending on the chosen GW model.

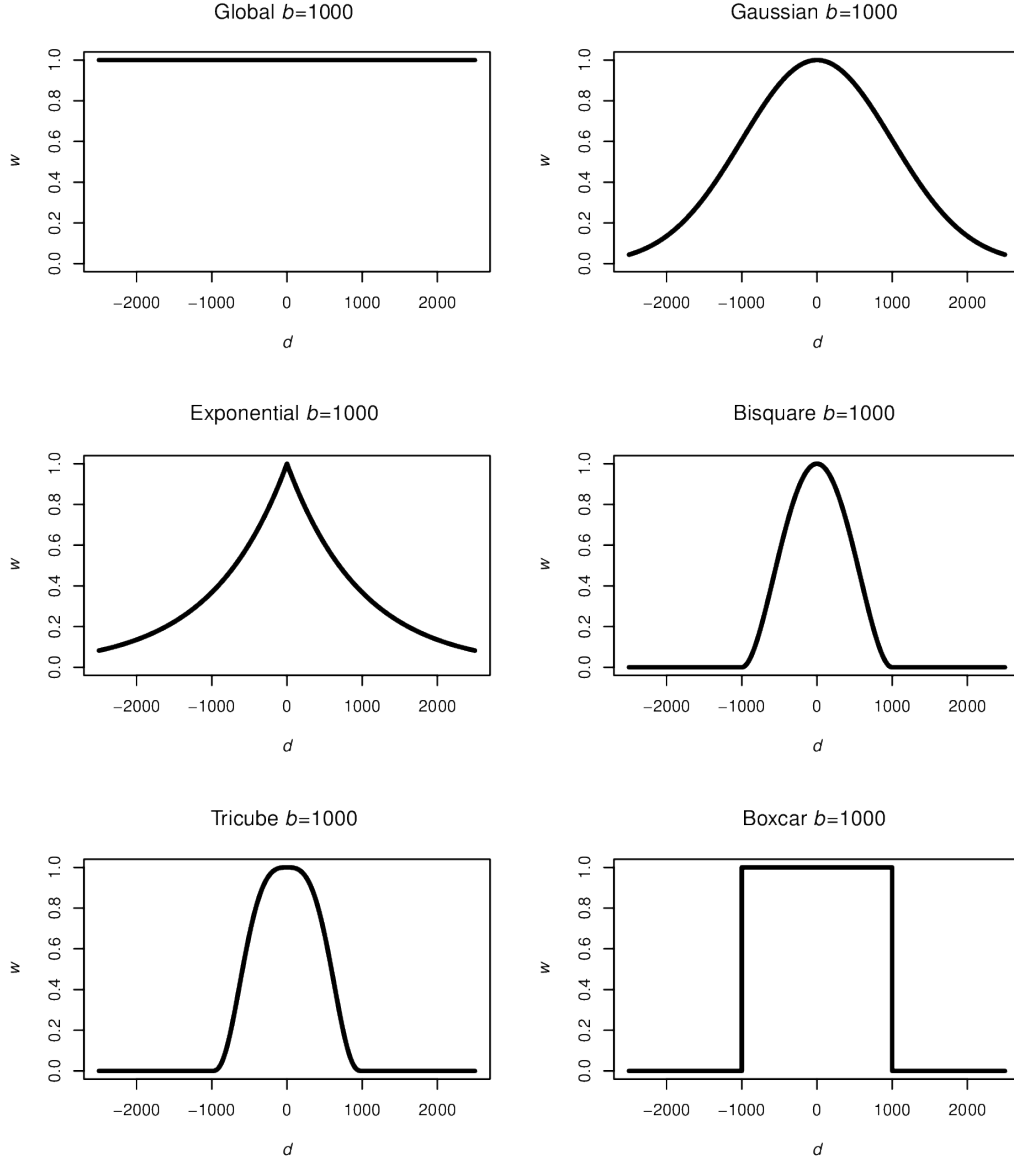


Figure 1: Plot of the six kernel functions, with the bandwidth $b = 1000$, and where w is the weight, and d is the distance between two observations.

3.3. Example

As an example, we find the distance matrix for the house price data for England and Wales (EWHP), as described in Section 2. Here, the distance matrix can be calculated: (a) within a function of a specific GW model or (b) outside of the function and saved using the function `gw.dist`. This flexibility is particularly useful for saving computation time when fitting several different GW models. Observe that we specify the Euclidean distance metric for this data. Other distance metrics could have been specified by: (1) modifying the parameter `p`, the power of the Minkowsky distance or (2) setting `longlat = TRUE` for the great circle distance. The

output of the function `gw.dist` is a matrix containing in each row the value of the diagonal of the distance matrix for each observation.

```
R> library("GWmodel")
R> data("EWHP")
R> houses.spdf <- SpatialPointsDataFrame(ewhp[, 1:2], ewhp)
R> houses.spdf[1:6, ]
```

	Easting	Northing	PurPrice	BldIntWr	BldPostW	Bld60s	Bld70s	Bld80s	TypDetch
1	599500	142200	65000	0	0	0	0	1	0
2	575400	167200	45000	0	0	0	0	0	0
3	530300	177300	50000	1	0	0	0	0	0
4	524100	170300	105000	0	0	0	0	0	0
5	426900	514600	175000	0	0	0	0	1	1
6	508000	190400	250000	0	1	0	0	0	1

	TypSemiD	TypFlat	FlrArea
1	1	0	78.94786
2	0	1	94.36591
3	0	0	41.33153
4	0	0	92.87983
5	0	0	200.52756
6	0	0	148.60773

```
R> DM <- gw.dist(dp.locat = coordinates(houses.spdf))
R> DM[1:7, 1:7]
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.00	34724.78	77592.848	80465.956	410454.0	103419.00	236725.0
[2,]	34724.78	0.00	46217.096	51393.579	377808.2	71281.13	202563.8
[3,]	77592.85	46217.10	0.000	9350.936	352792.9	25863.10	160741.1
[4,]	80465.96	51393.58	9350.936	0.000	357757.4	25753.06	160945.0
[5,]	410454.04	377808.17	352792.928	357757.362	0.0	334189.84	232275.4
[6,]	103419.00	71281.13	25863.101	25753.058	334189.8	0.00	135411.2
[7,]	236725.01	202563.77	160741.096	160945.022	232275.4	135411.23	0.0

4. GW summary statistics

This section presents the simplest form of GW modelling with GW summary statistics (Brunsdon *et al.* 2002; Fotheringham *et al.* 2002). Here, we describe how to calculate GW means, GW standard deviations and GW measures of skew; which constitute a set of basic GW summary statistics. To mitigate against any adverse effect of outliers on these local statistics, a set of robust alternatives are also described in GW medians, GW inter-quartile ranges and GW quantile imbalances. In addition, to such local *univariate* summary statistics, GW correlations are described in basic and robust forms (Pearson's and Spearman's, respectively); providing a set of local *bivariate* summary statistics.

Although fairly simple to calculate and map, GW summary statistics are considered a vital pre-cursor to an application of any subsequent GW model, such as a GW PCA (Section 5) or GW regression (Sections 6 to 8). For example, GW standard deviations (or GW inter-quartile ranges) will highlight areas of high variability for a given variable, areas where a subsequent application of a GW PCA or a GW regression may warrant close scrutiny. Basic and robust GW correlations provide a preliminary assessment of relationship non-stationarity between the dependent and an independent variable of a GW regression (Section 6). GW correlations also provide an assessment of local collinearity between two independent variables of a GW regression; which could then lead to the application of a locally compensated model (Section 7).

4.1. Basic GW summary statistics

For attributes z and y at any location i where w_{ij} accords to some kernel function of Section 3, definitions for a GW mean, a GW standard deviation, a GW measure of skew and a GW Pearson's correlation coefficient are respectively:

$$m(z_i) = \frac{\sum_{j=1}^n w_{ij} z_j}{\sum_{j=1}^n w_{ij}}$$

$$s(z_i) = \sqrt{\frac{\sum_{j=1}^n w_{ij} (z_j - m(z_i))^2}{\sum_{j=1}^n w_{ij}}}$$

$$b(z_i) = \sqrt[3]{\frac{\sum_{j=1}^n w_{ij} (z_j - m(z_i))^3}{\sum_{j=1}^n w_{ij} s(z_i)^3}}$$

and

$$\rho(z_i, y_i) = \frac{c(z_i, y_i)}{s(z_i)s(y_i)} \quad (1)$$

with the GW covariance:

$$c(z_i, y_i) = \frac{\sum_{j=1}^n w_{ij} [(z_j - m(z_i)) (y_j - m(y_i))]}{\sum_{j=1}^n w_{ij}}$$

4.2. Robust GW summary statistics

Definitions for a GW median, a GW inter-quartile range and a GW quantile imbalance, all require the calculation of GW quantiles at any location i ; the calculation of which are detailed in [Brunsdon *et al.* \(2002\)](#). Thus if we calculate GW quartiles, the GW median is the second GW quartile; and the GW inter-quartile range is the third minus the first GW quartile. The GW quantile imbalance measures the symmetry of the middle part of the local distribution and is based on the position of the GW median relative to the first and third GW quartiles. It ranges from -1 (when the median is very close to the first GW quartile) to 1 (when the median is very close to the third GW quartile), and is zero if the median bisects the first and third GW quartiles. To find a GW Spearman's correlation coefficient, the local data for z and for y , each need to be ranked using the same approach as that used to calculate the GW quantiles. The locally ranked variables are then simply fed into Equation 1.

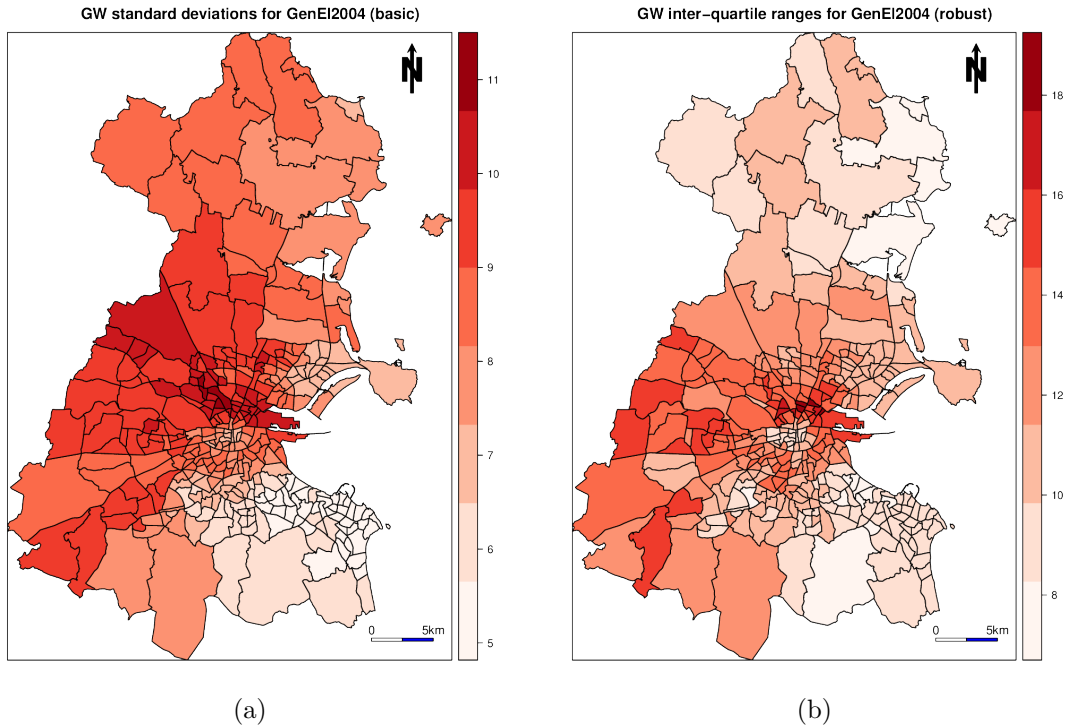


Figure 2: (a) Basic and (b) robust GW measures of variability for **GenEl2004** (turnout).

4.3. Example

For demonstration of basic and robust GW summary statistics, we use the Dublin voter turnout data. Here we investigate the local variability in voter turnout (**GenEl2004**), which is the dependent variable in the regressions of Sections 6 and 7. We also investigate the local relationships between: (i) turnout and **LARent** and (ii) **LARent** and **Unempl** (i.e., two independent variables in the regressions of Sections 6 and 7).

For any GW model calibration, it is prudent to experiment with different kernel functions. For our chosen GW summary statistics, we specify box-car and bi-square kernels; where the former relates to an un-weighted moving window, whilst the latter relates to a weighted one (from Section 3). GW models using box-car kernels are useful in that the identification of outlying relationships or structures are more likely (Lloyd and Shuttleworth 2005; Harris and Brunsdon 2010). Such calibrations more easily relate to the global model form (see Section 7) and in turn, tend to provide an intuitive understanding of the degree of heterogeneity in the process. Observe that it is always possible that the spatial process is essentially homogeneous, and in such cases, the output of a GW model can confirm this.

The spatial arrangement of the EDs in Greater Dublin is not a tessellation of equally sized zones, so it makes sense to specify an adaptive kernel bandwidth. For example, if we specify a bandwidth of $N = 100$, the box-car and bi-square kernels will change in radius but will always include the closest 100 EDs for each local summary statistic. We also user-specify the bandwidths. Note that bandwidths for GW means or medians can be found optimally via cross-validation, as an objective function exists. That is, ‘leave-one-out’ predictions can be found and compared to the actual data, across a range of bandwidths. The optimal band-

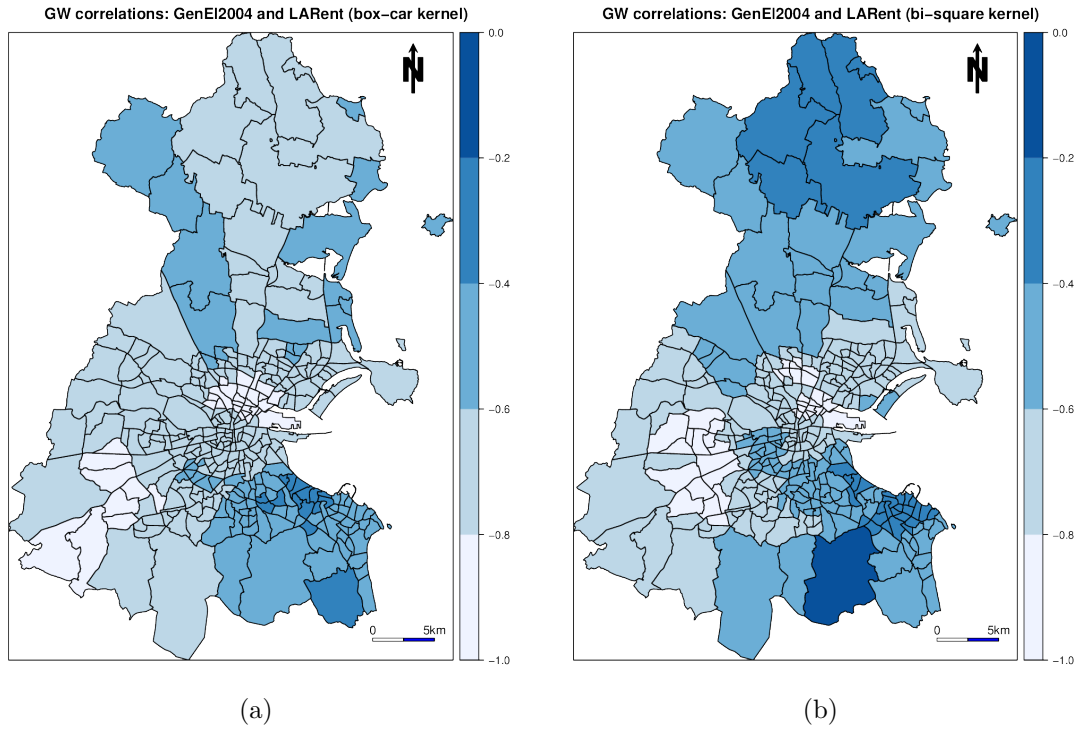


Figure 3: (a) Box-car and (b) bi-square specified GW correlations for **GenEl2004** and **LARent**.

width is that which provides the most accurate predictions. However, such bandwidth selection functions are not yet incorporated in **GWmodel**. For all other GW summary statistics, bandwidths can only be user-specified, as no objective function exists for them.

Commands to conduct our local analysis are as follows, where we use the function **gwss** with two different specifications to find our GW summary statistics. We specify box-car and bi-square kernels, each with an adaptive bandwidth of $N = 48$ (approximately 15% of the data). To find robust GW summary statistics based on quantiles, the **gwss** function is specified with **quantiles = TRUE** (observe that we do not need to do this for our robust GW correlations).

```
R> data("DubVoter")
R> gw.ss.bx <- gwss(Dub.voter, vars = c("GenEl2004", "LARent", "Unempl"),
+   kernel = "boxcar", adaptive = TRUE, bw = 48, quantile = TRUE)
R> gw.ss.bs <- gwss(Dub.voter, vars = c("GenEl2004", "LARent", "Unempl"),
+   kernel = "bisquare", adaptive = TRUE, bw = 48)
```

From these calibrations, we present three pairs of example visualizations: (a) basic and robust GW measures of variability for **GenEl2004** (each using a box-car kernel) in Figure 2; (b) box-car and bi-square specified (basic) GW correlations for **GenEl2004** and **LARent** in Figure 3; and (c) basic and robust GW correlations for **LARent** and **Unempl** (each using a bi-square kernel) in Figure 4. Commands to conduct these visualizations (using palettes from **RColorBrewer**, [Neuwirth 2011](#)), are as follows:

```
R> library("RColorBrewer")
R> map.na = list("SpatialPolygonsRescale", layout.north.arrow(),
```

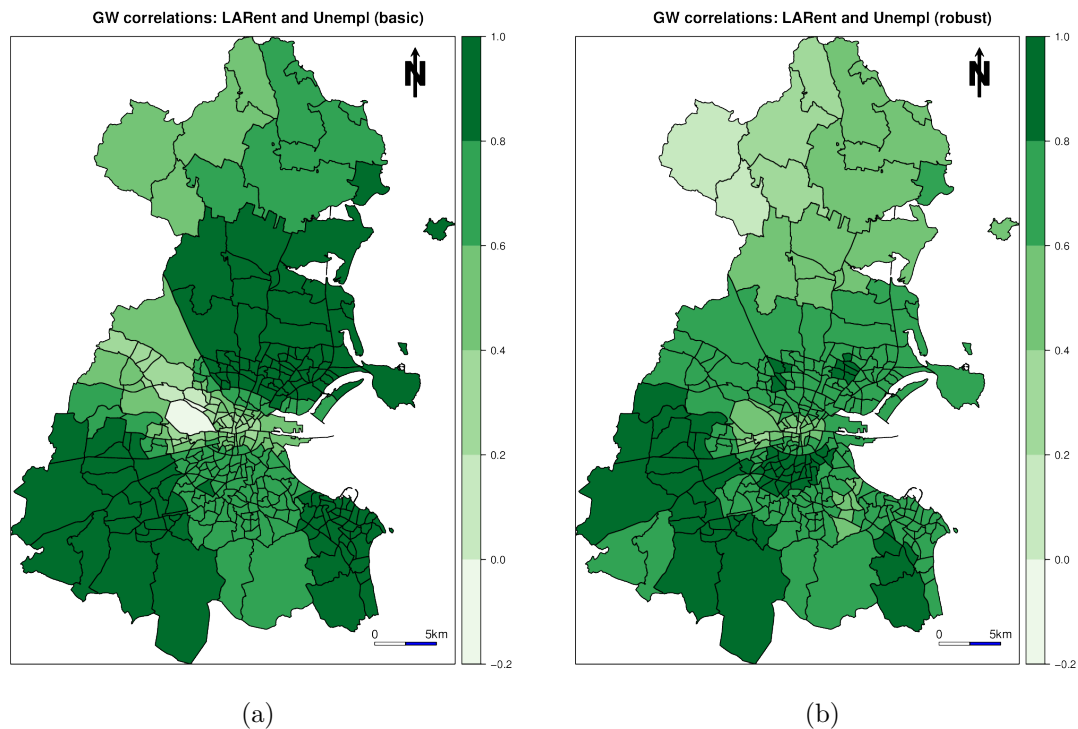


Figure 4: (a) Basic and (b) robust GW correlations for LARent and Unempl.

```
+ offset = c(329000, 261500), scale = 4000, col = 1)
R> map.scale.1 = list("SpatialPolygonsRescale", layout.scale.bar(),
+ offset = c(326500, 217000), scale = 5000, col = 1,
+ fill = c("transparent", "blue"))
R> map.scale.2 = list("sp.text", c(326500, 217900), "0", cex = 0.9, col = 1)
R> map.scale.3 = list("sp.text", c(331500, 217900), "5km", cex = 0.9, col = 1)
R> map.layout <- list(map.na, map.scale.1, map.scale.2, map.scale.3)
R> mypalette.1 <- brewer.pal(8, "Reds")
R> mypalette.2 <- brewer.pal(5, "Blues")
R> mypalette.3 <- brewer.pal(6, "Greens")
R> spplot(gw.ss.bx$SDF, "GenEl2004_LSD", key.space = "right",
+ col.regions = mypalette.1, cuts = 7, sp.layout = map.layout,
+ main = "GW standard deviations for GenEl2004 (basic)")
R> spplot(gw.ss.bx$SDF, "GenEl2004_IQR", key.space = "right",
+ col.regions = mypalette.1, cuts = 7, sp.layout = map.layout,
+ main = "GW inter-quartile ranges for GenEl2004 (robust)")
R> spplot(gw.ss.bx$SDF, "Corr_GenEl2004.LARent", key.space = "right",
+ col.regions = mypalette.2, at = c(-1, -0.8, -0.6, -0.4, -0.2, 0),
+ main = "GW correlations: GenEl2004 and LARent (box-car kernel)",
+ sp.layout = map.layout)
R> spplot(gw.ss.bs$SDF, "Corr_GenEl2004.LARent", key.space = "right",
+ col.regions = mypalette.2, at = c(-1, -0.8, -0.6, -0.4, -0.2, 0),
+ main = "GW correlations: GenEl2004 and LARent (bi-square kernel)",
```

```

+   sp.layout = map.layout)
R> spplot(gw.ss.bs$SDF, "Corr_LARent.Unempl", key.space = "right",
+   col.regions = mypalette.3, at = c(-0.2, 0, 0.2, 0.4, 0.6, 0.8, 1),
+   main = "GW correlations: LARent and Unempl (basic)",
+   sp.layout = map.layout)
R> spplot(gw.ss.bs$SDF, "Spearman_rho_LARent.Unempl", key.space = "right",
+   col.regions = mypalette.3, at = c(-0.2, 0, 0.2, 0.4, 0.6, 0.8, 1),
+   main = "GW correlations: LARent and Unempl (robust)",
+   sp.layout = map.layout)

```

From Figure 2, we can see that turnout appears highly variable in areas of central and west Dublin. From Figure 3, the relationship between turnout and LARent appears non-stationary, where this relationship is strongest in areas of central and south-west Dublin. Here turnout tends to be low while local authority renting tends to be high. From Figure 4, consistently strong positive correlations between LARent and Unempl are found in south-west Dublin. This is precisely an area of Dublin where local collinearity in the GW regression of Section 7 is found to be strong and a cause for concern.

From these visualizations, it is clearly important to experiment with the calibration of a GW model, as subtle differences in our perception of the non-stationary effect can result by a simple altering of the specification. Experimentation with different bandwidth sizes is also important, especially in cases when an optimal bandwidth cannot be specified. Observe that all GW models are primarily viewed as exploratory spatial data analysis (ESDA) tools and as such, experimentation is a vital aspect of this.

5. GW principal components analysis

Principal components analysis (PCA) is a key method for the analysis of multivariate data (see Jolliffe 2002). A member of the unconstrained ordination family, it is commonly used to explain the covariance structure of a (high-dimensional) multivariate data set using only a few components (i.e., provide a low-dimensional alternative). The components are linear combinations of the original variables and can potentially provide a better understanding of differing sources of variation and structure in the data. These may be visualized and interpreted using associated graphics. In geographical settings, standard PCA, in which the components do not depend on location, may be replaced with a GW PCA (Fotheringham *et al.* 2002; Lloyd 2010a; Harris *et al.* 2011a), to account for spatial heterogeneity in the structure of the multivariate data. In doing so, GW PCA can identify regions where assuming the same underlying structure in all locations is inappropriate or over-simplistic. GW PCA can assess: (i) how (effective) data dimensionality varies spatially and (ii) how the original variables influence each spatially-varying component. In part, GW PCA resembles the bivariate GW correlations of Section 4 in a multivariate sense, as both are unlike the multivariate GW regressions of Sections 6 to 8, since there is no distinction between dependent and independent variables. Key challenges in GW PCA are: (a) finding the scale at which each localized PCA should operate and (b) visualising and interpreting the output that results from its application. As with any GW model, GW PCA is constructed using weighted data that is controlled by the kernel function and its bandwidth (Section 3).

5.1. GW PCA

More formally, for a vector of observed variables x_i at spatial location i with coordinates (u, v) , GW PCA involves regarding x_i as conditional on u and v , and making the mean vector μ and covariance matrix Σ , functions of u and v . That is, $\mu(u, v)$ and $\Sigma(u, v)$ are the local mean vector and the local covariance matrix, respectively. To find the local principal components, the decomposition of the local covariance matrix provides the local eigenvalues and local eigenvectors. The product of the i -th row of the data matrix with the local eigenvectors for the i -th location provides the i -th row of local component scores. The local covariance matrix is:

$$\Sigma(u, v) = X^\top W(u, v) X$$

where X is the data matrix (with n rows for the observations and m columns for the variables); and $W(u, v)$ is a diagonal matrix of geographic weights. The local principal components at location (u_i, v_i) can be written as:

$$L(u_i, v_i) V(u_i, v_i) L(u_i, v_i)^\top = \Sigma(u_i, v_i)$$

where $L(u_i, v_i)$ is a matrix of local eigenvectors; $V(u_i, v_i)$ is a diagonal matrix of local eigenvalues; and $\Sigma(u_i, v_i)$ is the local covariance matrix. Thus for a GW PCA with m variables, there are m components, m eigenvalues, m sets of component scores, and m sets of component loadings at each observed location. We can also obtain eigenvalues and their associated eigenvectors at un-observed locations, although as no data exists for these locations, we cannot obtain component scores.

5.2. Robust GW PCA

A robust GW PCA can also be specified, so as to reduce the effect of anomalous observations on its outputs. Outliers can artificially increase local variability and mask key features in local data structures. To provide a robust GW PCA, each local covariance matrix is estimated using the robust minimum covariance determinant (MCD) estimator ([Rousseeuw 1985](#)). The MCD estimator searches for a subset of h data points that has the smallest determinant for their basic sample covariance matrix. Crucial to the robustness and efficiency of this estimator is h , and we specify a default value of $h = 0.75n$, following the recommendation of [Varmuza and Filzmoser \(2009\)](#).

5.3. Example

For applications of PCA and GW PCA, we again use the Dublin voter turnout data, this time focussing on the eight variables: `DiffAdd`, `LARent`, `SC1`, `Unempl`, `LowEduc`, `Age18_24`, `Age25_44` and `Age45_64` (i.e., the independent variables of the regression fits in [Sections 6 and 7](#)). Although measured on the same scale, the variables are not of a similar magnitude. Thus, we standardize the data and specify our PCA with the covariance matrix. The same (globally) standardized data is also used in our GW PCA calibrations, which are similarly specified with (local) covariance matrices. The effect of this standardization is to make each variable have equal importance in the subsequent analysis (at least for the PCA case)². The

²The use of un-standardized data, or the use of locally-standardized data with GW PCA is a subject of current investigation.

basic and robust PCA results are found using `scale`, `princomp` and `covMcd` functions, as follows:

```
R> Data.scaled <- scale(as.matrix(Dub.voter@data[, 4:11]))
R> pca.basic <- princomp(Data.scaled, cor = FALSE)
R> (pca.basic$sdev^2 / sum(pca.basic$sdev^2)) * 100
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7
	36.084435	25.586984	11.919681	10.530373	6.890565	3.679812	3.111449
	Comp.8						
	2.196701						

```
R> pca.basic$loadings
```

Loadings:

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
DiffAdd	0.389	-0.444		-0.149	0.123	0.293	0.445	0.575
LARent	0.441	0.226	0.144	0.172	0.612	0.149	-0.539	0.132
SC1	-0.130	-0.576		-0.135	0.590	-0.343		-0.401
Unempl	0.361	0.462		0.189	0.197		0.670	-0.355
LowEduc	0.131	0.308	-0.362	-0.861				
Age18_24	0.237		0.845	-0.359	-0.224			-0.200
Age25_44	0.436	-0.302	-0.317		-0.291	0.448	-0.177	-0.546
Age45_64	-0.493	0.118	0.179	-0.144	0.289	0.748	0.142	-0.164

```
R> R.COV <- covMcd(Data.scaled, cor = FALSE, alpha = 0.75)
R> pca.robust <- princomp(Data.scaled, covmat = R.COV, cor = FALSE)
R> pca.robust$sdev^2 / sum(pca.robust$sdev^2)
```

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
	0.419129445	0.326148321	0.117146840	0.055922308	0.043299600
	Comp.6	Comp.7	Comp.8		
	0.017251964	0.014734597	0.006366926		

```
R> pca.robust$loadings
```

Loadings:

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
DiffAdd	0.512		-0.180	0.284	-0.431	0.659		
LARent		-0.139		0.310	0.119			-0.932
SC1	0.559	0.591	0.121	0.368	0.284	-0.324		
Unempl	-0.188	-0.394		0.691		-0.201	0.442	0.307
LowEduc	-0.102	-0.186		0.359			-0.895	0.149
Age18_24			-0.937		0.330			
Age25_44	0.480	-0.437		-0.211	-0.407	-0.598		
Age45_64	-0.380	0.497	-0.264	0.178	-0.665	-0.241		

From the ‘percentage of total variance’ (PTV) results, the first three components collectively account for 73.6% and 86.2% of the variation in the data, for the basic and robust PCA, respectively. From the tables of loadings, component one would appear to represent older residents (Age45_64) in the basic PCA or represent affluent residents (SC1) in the robust PCA. Component two, appears to represent affluent residents in both the basic and robust PCA. These are whole-map statistics (Openshaw, Charlton, Wymer, and Craft 1987) and interpretations that represent a Dublin-wide average. However, it is possible that they do not represent local social structure particularly reliably. If this is the case, an application of GW PCA may be useful, which will now be demonstrated.

Kernel bandwidths for GW PCA can be found automatically using a cross-validation approach, similar in nature to that used in GW regression (Section 6). Details of this automated procedure are described in Harris *et al.* (2011a), where, a ‘leave-one-out’ cross-validation (CV) score is computed for all possible bandwidths and an optimal bandwidth relates to the smallest CV score found. With this procedure, it is currently necessary to decide *a priori* upon the number of components to retain (k , say), and a different optimal bandwidth results for each k . The procedure does not yield an optimal bandwidth if all components are retained (i.e., $m = k$); in this case, the bandwidth must be user-specified. Thus for our analysis, an optimal adaptive bandwidth is found using a bi-square kernel, for both a basic and a robust GW PCA. Here, $k = 3$ is chosen on an *a priori* basis. With **GWmodel**, the `bw.gwpca` function is used in the following set of commands, where the standardized data is converted to a spatial form via the `SpatialPointsDataFrame` function.

```
R> Coords <- as.matrix(cbind(Dub.voter$X, Dub.voter$Y))
R> Data.scaled.spdf <- SpatialPointsDataFrame(Coords,
+   as.data.frame(Data.scaled))
R> bw.gwpca.basic <- bw.gwpca(Data.scaled.spdf, vars = colnames(
+   Data.scaled.spdf@data), k = 3, robust = FALSE, adaptive = TRUE)
R> bw.gwpca.basic
```

```
[1] 131
```

```
R> bw.gwpca.robust <- bw.gwpca(Data.scaled.spdf, vars = colnames(
+   Data.scaled.spdf@data), k = 3, robust = TRUE, adaptive = TRUE)
R> bw.gwpca.robust
```

```
[1] 130
```

Inspecting the values of `bw.gwpca.basic` and `bw.gwpca.robust` show that (very similar) optimal bandwidths of $N = 131$ and $N = 130$ will be used to calibrate the respective basic and robust GW PCA fits. Observe that we now specify all $k = 8$ components, but will focus our investigations on only the first three components. This specification ensures that the variation locally accounted for by each component, is estimated correctly. The two GW PCA fits are found using the `gwpca` function as follows:

```
R> gwpca.basic <- gwpca(Data.scaled.spdf,
+   vars = colnames(Data.scaled.spdf@data), bw = bw.gwpca.basic, k = 8,
+   robust = FALSE, adaptive = TRUE)
```

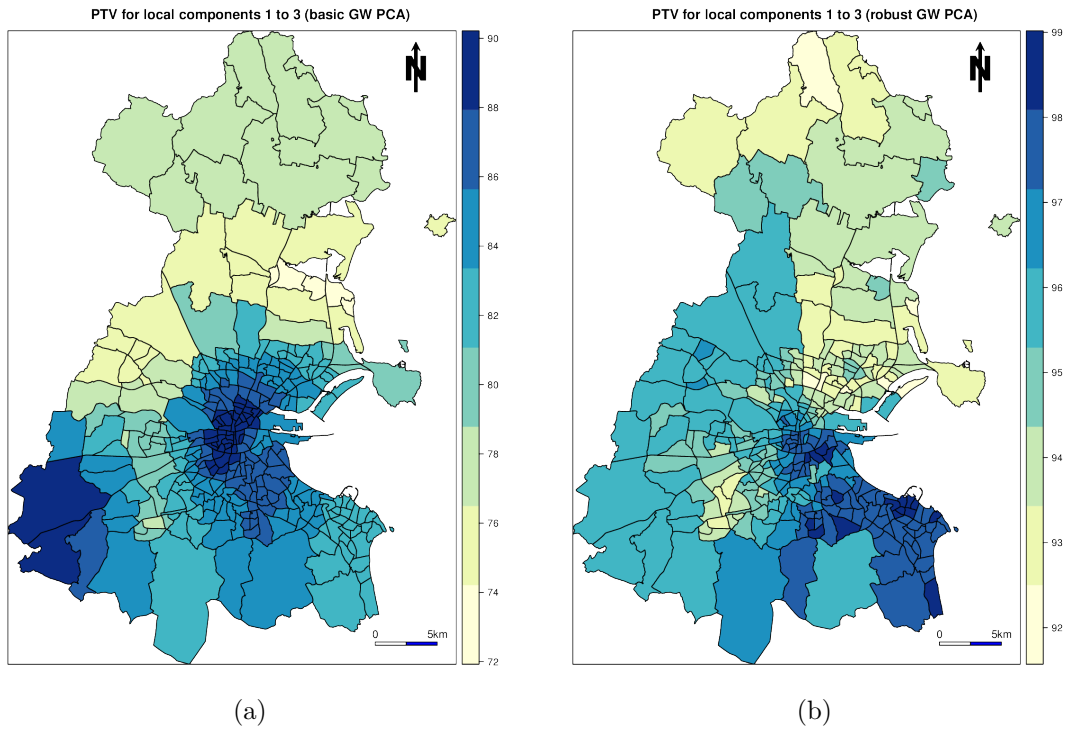



Figure 5: (a) Basic and (b) robust PTV data for the first three local components.

```
R> gwpca.robust <- gwpca(Data.scaled.spdf,
+   vars = colnames(Data.scaled.spdf@data), bw = bw.gwpca.robust, k = 8,
+   robust = TRUE, adaptive = TRUE)
```

The GW PCA outputs³ may now be visualized and interpreted, focusing on: (1) how data dimensionality varies spatially and (2) how the original variables influence the components. For the former, the spatial distribution of local PTV for say, the first three components can be mapped. Commands to conduct this mapping for basic and robust GW PCA outputs are as follows, where the `prop.var` function is used to find the PTV data, which is then added to the `Dub.voter` spatial data frame, so that it can be easily mapped using the `spplot` function.

```
R> prop.var <- function(gwpca.obj, n.components) {
+   return((rowSums(gwpca.obj$var[, 1:n.components]) /
+     rowSums(gwpca.obj$var)) * 100)
+ }
R> var.gwpca.basic <- prop.var(gwpca.basic, 3)
R> var.gwpca.robust <- prop.var(gwpca.robust, 3)
R> Dub.voter$var.gwpca.basic <- var.gwpca.basic
R> Dub.voter$var.gwpca.robust <- var.gwpca.robust
R> mypalette.4 <- brewer.pal(8, "YlGnBu")
R> spplot(Dub.voter, "var.gwpca.basic", key.space = "right",
+   col.regions = mypalette.4, cuts = 7, sp.layout = map.layout,
```

³For a more objective and direct comparison of the basic and robust fits, the use of the same bandwidth (say that found optimally for the basic fit) may be preferable.

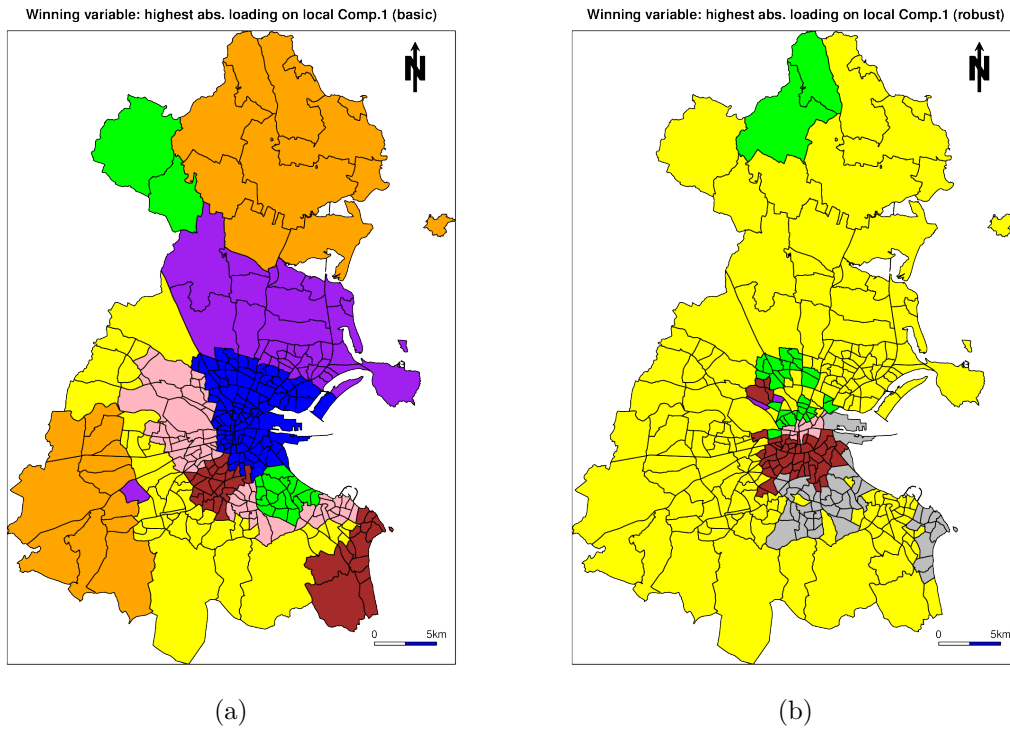


Figure 6: (a) Basic and (b) robust GW PCA results for the winning variable on the first component. Map legends are: DiffAdd - light pink; LARent - blue; SC1 - grey; Unempl - purple; LowEduc - orange; Age18_24 - green; Age25_44 - brown; and Age45_64 - yellow.

```
+ main = "PTV for local components 1 to 3 (basic GW PCA)"
R> spplot(Dub.voter, "var.gwpca.robust", key.space = "right",
+ col.regions = mypalette.4, cuts = 7, sp.layout = map.layout,
+ main = "PTV for local components 1 to 3 (robust GW PCA)")
```

Figure 5 presents the local PTV maps for the two GW PCA fits. There is clear geographical variation in the PTV data and a higher PTV is generally accounted for in the local case, than in the global case. The spatial patterns in both maps are broadly similar, with higher percentages located in the south, whilst lower percentages are located in the north. As would be expected, the robust PTV data is consistently higher than the basic PTV data. Variation in the basic PTV data is also greater than that found in the robust PTV data. Large (relative) differences between the basic and robust PTV outputs (e.g., in south-west Dublin) can be taken to indicate the existence of global or possibly, local multivariate outliers.

We can next visualize how each of the eight variables locally influence a given component, by mapping the ‘winning variable’ with the highest absolute loading. For brevity, we present such maps for the first component, only. Commands to conduct this mapping for basic and robust GW PCA outputs are as follows:

```
R> loadings.pc1.basic <- gwpca.basic$loadings[, , 1]
R> win.item.basic = max.col(abs(loadings.pc1.basic))
R> loadings.pc1.robust <- gwpca.robust$loadings[, , 1]
R> win.item.robust = max.col(abs(loadings.pc1.robust))
```

```

R> Dub.voter$win.item.basic <- win.item.basic
R> Dub.voter$win.item.robust <- win.item.robust
R> mypalette.5 <- c("lightpink", "blue", "grey", "purple", "orange",
+   "green", "brown", "yellow")
R> spplot(Dub.voter, "win.item.basic", key.space = "right",
+   col.regions = mypalette.5, at = c(1, 2, 3, 4, 5, 6, 7, 8, 9),
+   main = "Winning variable: highest abs. loading on local Comp.1 (basic)",
+   colorkey = FALSE, sp.layout = map.layout)
R> spplot(Dub.voter, "win.item.robust", key.space = "right",
+   col.regions = mypalette.5, at = c(1, 2, 3, 4, 5, 6, 7, 8, 9),
+   main = "Winning variable: highest abs. loading on local Comp.1 (robust)",
+   colorkey = FALSE, sp.layout = map.layout)

```

Figure 6 presents the ‘winning variable’ maps for the two GW PCA fits, where we can observe clear geographical variation in the influence of each variable on the first component. For basic GW PCA, low educational attainment (Low_Educ) dominates in the northern and south-western EDs, whilst public housing (LARent) dominates in the EDs of central Dublin. The corresponding PCA ‘winning variable’ is Age45_64, which is clearly not dominant throughout Dublin. Variation in the results from basic GW PCA is much greater than that found with robust GW PCA (reflecting analogous results to that found with the PTV data). For robust GW PCA, Age45_64 does in fact dominate in most areas, thus reflecting a closer correspondence to the global case - but interestingly only the basic fit, and not the robust fit.

6. GW regression

6.1. Basic GW regression

The most popular GW model is GW regression (Brunsdon *et al.* 1996, 1998), where spatially-varying relationships are explored between the dependent and independent variables. Exploration commonly consists of mapping the resultant local regression coefficient estimates and associated (pseudo) t -values to determine evidence of non-stationarity. The basic form of the GW regression model is:

$$y_i = \beta_{i0} + \sum_{k=1}^m \beta_{ik} x_{ik} + \epsilon_i$$

where y_i is the dependent variable at location i ; x_{ik} is the value of the k th independent variable at location i ; m is the number of independent variables; β_{i0} is the intercept parameter at location i ; β_{ik} is the local regression coefficient for the k th independent variable at location i ; and ϵ_i is the random error at location i .

As data are geographically weighted, nearer observations have more influence in estimating the local set of regression coefficients than observations farther away. The model measures the inherent relationships around each regression point i , where each set of regression coefficients is estimated by a weighted least squares approach. The matrix expression for this estimation is:

$$\hat{\beta}_i = \left(X^\top W(u_i, v_i) X \right)^{-1} X^\top W(u_i, v_i) y$$

where X is the matrix of the independent variables with a column of 1s for the intercept; y is the dependent variable vector; $\hat{\beta}_i = (\beta_{i0}, \dots, \beta_{im})^\top$ is the vector of $m + 1$ local regression coefficients; and W_i is the diagonal matrix denoting the geographical weighting of each observed data for regression point i at location (u_i, v_i) . This weighting is determined by some kernel function as described in Section 3.

An optimum kernel bandwidth for GW regression can be found by minimising some model fit diagnostic, such as a leave-one-out cross-validation (CV) score (Bowman 1984), which only accounts for model prediction accuracy; or the Akaike Information Criterion (AIC) (Akaike 1973), which accounts for model parsimony (i.e., a trade-off between prediction accuracy and complexity). In practice, a corrected version of the AIC is used, which unlike basic AIC is a function of sample size (Hurvich, Simonoff, and Tsai 1998). For GW regression, this entails that fits using small bandwidths receive a higher penalty (i.e., are more complex) than those using large bandwidths. Thus for a GW regression with a bandwidth b , its AICc can be found from:

$$\text{AIC}_c(b) = 2n \ln(\hat{\sigma}) + n \ln(2\pi) + n \left\{ \frac{n + \text{tr}(S)}{n - 2 - \text{tr}(S)} \right\}$$

where n is the (local) sample size (according to b); $\hat{\sigma}$ is the estimated standard deviation of the error term; and $\text{tr}(S)$ denotes the trace of the hat matrix S . The hat matrix is the projection matrix from the observed y to the fitted values, \hat{y} .

6.2. Robust GW regression

To identify and reduce the effect of outliers in GW regression, various robust extensions have been proposed, two of which are described in Fotheringham *et al.* (2002). The first robust model re-fits a GW regression with a filtered data set that has been found by removing observations that correspond to large externally studentized residuals of an initial GW regression fit. An externally studentized residual for each regression location i is defined as:

$$r_i = \frac{e_i}{\hat{\sigma}_{-i} \sqrt{q_{ii}}}$$

where e_i is the residual at location i ; $\hat{\sigma}_{-i}$ is a leave-one-out estimate of $\hat{\sigma}$; and q_{ii} is the i th element of $(I - S)(I - S)^\top$. Observations are deemed outlying and filtered from the data if they have $|r_i| > 3$. The second robust model, iteratively down-weights observations that correspond to large residuals. This (non-geographical) weighting function w_r on the residual e_i is typically taken as:

$$w_r(e_i) = \begin{cases} 1, & \text{if } |e_i| \leq 2\hat{\sigma} \\ [1 - (|e_i| - 2)^2]^2, & \text{if } 2\hat{\sigma} < |e_i| < 3\hat{\sigma} \\ 0 & \text{otherwise} \end{cases}$$

Observe that both approaches have an element of subjectivity, where the filtered data approach depends on the chosen residual cut-off (in this case, 3) and the iterative (automatic) approach depends on the chosen down-weighting function, with its associated cut-offs.

6.3. Example

We now demonstrate the fitting of the basic and robust GW regressions described, to the Dublin voter turnout data. Our regressions attempt to accurately predict the proportion of

the electorate who turned out on voting night to cast their vote in the 2004 General Election in Ireland. The dependent variable is `GenEl2004` and the eight independent variables are `DiffAdd`, `LARent`, `SC1`, `Unempl`, `LowEduc`, `Age18_24`, `Age25_44` and `Age45_64`.

A global correlation analysis suggests that voter turnout is negatively associated with the independent variables, except for social class (`SC1`) and older adults (`Age45_64`). Public renters (`LARent`) and unemployed (`Unempl`) have the highest correlations (both negative), in this respect. The GW correlation analysis from Section 4 indicates that some of these relationships are non-stationary. The global regression fit to this data yields an R-squared value of 0.63 and details of this fit can be summarized as follows:

```
R> lm.global <- lm(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl + LowEduc +
+   Age18_24 + Age25_44 + Age45_64, data = Dub.voter)
R> summary(lm.global)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	77.70467	3.93928	19.726	< 2e-16	***
DiffAdd	-0.08583	0.08594	-0.999	0.3187	
LARent	-0.09402	0.01765	-5.326	1.92e-07	***
SC1	0.08637	0.07085	1.219	0.2238	
Unempl	-0.72162	0.09387	-7.687	1.96e-13	***
LowEduc	-0.13073	0.43022	-0.304	0.7614	
Age18_24	-0.13992	0.05480	-2.554	0.0111	*
Age25_44	-0.35365	0.07450	-4.747	3.15e-06	***
Age45_64	-0.09202	0.09023	-1.020	0.3086	

Next, we conduct a model specification exercise in order to help find an independent variable subset for our basic GW regression. As an aide to this task, a pseudo stepwise procedure is used that proceeds in a forward direction. The procedure can be described in the following four steps, where the results are visualized using associated plots of each model's AIC_c values:

1. Start by calibrating all possible bivariate GW regressions by sequentially regressing a single independent variable against the dependent variable;
2. Find the best performing model which produces the minimum AIC_c , and permanently include the corresponding independent variable in subsequent models;
3. Sequentially introduce a variable from the remaining group of independent variables to construct new models with the permanently included independent variables, and determine the next permanently included variable from the best fitting model that has the minimum AIC_c ;
4. Repeat step 3 until all independent variables are permanently included in the model.

The function to perform this procedure is `model.selection.gwr`, whose AIC_c outputs are sorted using `model.sort.gwr` and then inputted to `model.view.gwr` to provide a useful visualization of the AIC_c data (see Figure 7). This approach can be conducted in a rudimentary form, where the bandwidth is user-specified beforehand and remains the same for

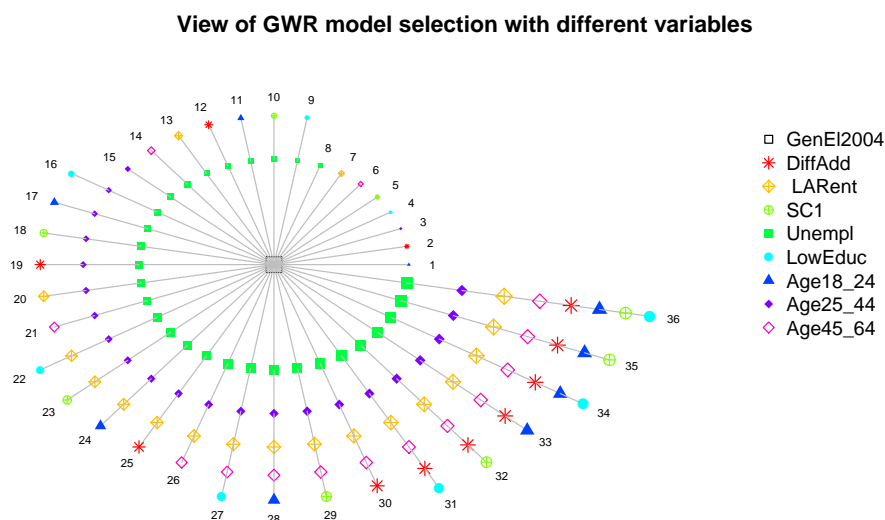


Figure 7: Model view of the stepwise specification procedure.

each GW regression fit. Alternatively, a more refined model specification exercise enables the re-calculation of an optimal bandwidth for each GW regression fit. As a demonstration, a rudimentary specification is conducted, by running the following sequence of commands. Observe that a bi-square kernel is specified with a user-specified adaptive bandwidth of $N = 80$.

```
R> DeVar <- "GenEl2004"
R> InDeVars <- c("DiffAdd", "LARent", "SC1", "Unempl", "LowEduc",
+   "Age18_24", "Age25_44", "Age45_64")
R> model.sel <- model.selection.gwr(DeVar, InDeVars, data = Dub.voter,
+   kernel = "bisquare", adaptive = TRUE, bw = 80)
R> sorted.models <- model.sort.gwr(model.sel, numVars = length(InDeVars),
+   ruler.vector = model.sel[[2]][,2])
R> model.list <- sorted.models[[1]]
R> model.view.gwr(DeVar, InDeVars, model.list = model.list)
R> plot(sorted.models[[2]][,2], col = "black", pch = 20, lty = 5,
+   main = "Alternative view of GWR model selection procedure",
+   ylab = "AICc", xlab = "Model number", type = "b")
```

Figure 7 presents a circle view of the 36 GW regressions (numbered 1 to 36) that result from this stepwise procedure. Here the dependent variable is located in the centre of the chart and the independent variables are represented as nodes differentiated by shapes and colors. The first independent variable that is permanently included is `Unempl`, the second is `Age25_44`, and the last is `LowEduc`. Figure 8 displays the corresponding AIC_c values from the same fits of Figure 7. The two graphs work together, explaining model performance when more and more variables are introduced. Clearly, AIC_c values continue to fall until all independent variables are included. Results suggest that continuing with all eight independent variables is worthwhile (at least for our user-specified bandwidth).

We can now proceed to the correct calibration of our chosen GW regression specification. Here, we find its true (i.e., optimal) bandwidth using the function `bw.gwr` and then use this

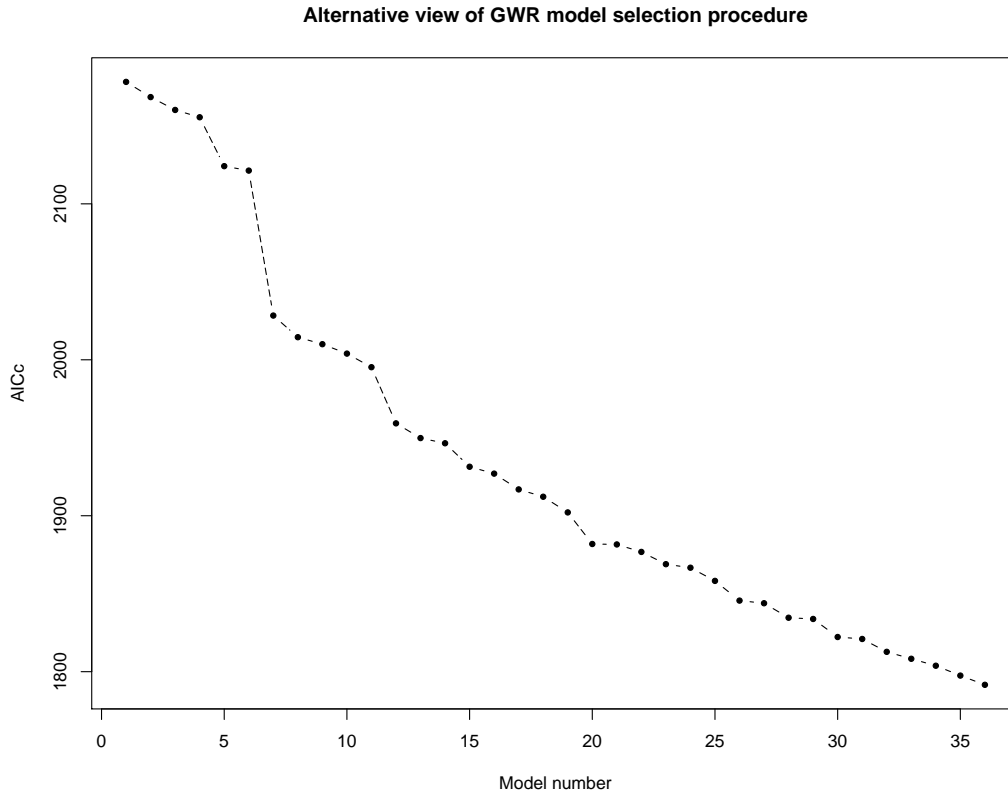


Figure 8: AIC_c values for the same 36 GW regressions of Figure 7.

bandwidth to parametrize the same GW regression with the function `gwr.basic`. The optimal bandwidth is found at $N = 109$. Commands for these operations are as follows, where the print function provides a useful report of the global and GW regression fits, with summaries of their regression coefficients, diagnostic information and F-test results (following [Leung *et al.* 2000](#)). The report is designed to match the output of the GW regression v3.0 executable software [Charlton *et al.* \(2003\)](#).

```
R> bw.gwr.1 <- bw.gwr(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter,
+   approach = "AICc", kernel = "bisquare", adaptive = TRUE)
R> bw.gwr.1
```

```
[1] 109
```

```
R> gwr.res <- gwr.basic(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter,
+   bw = bw.gwr.1, kernel = "bisquare", adaptive = TRUE, F123.test = TRUE)
R> print(gwr.res)
```

To map the GW regression outputs, the following commands can be used to each field of spatial data frame object `gwr.res$SDF`. As an example, we map the coefficient estimates for `LowEduc` in Figure 9a, where this variable's relationship to voter turnout has clear geographical

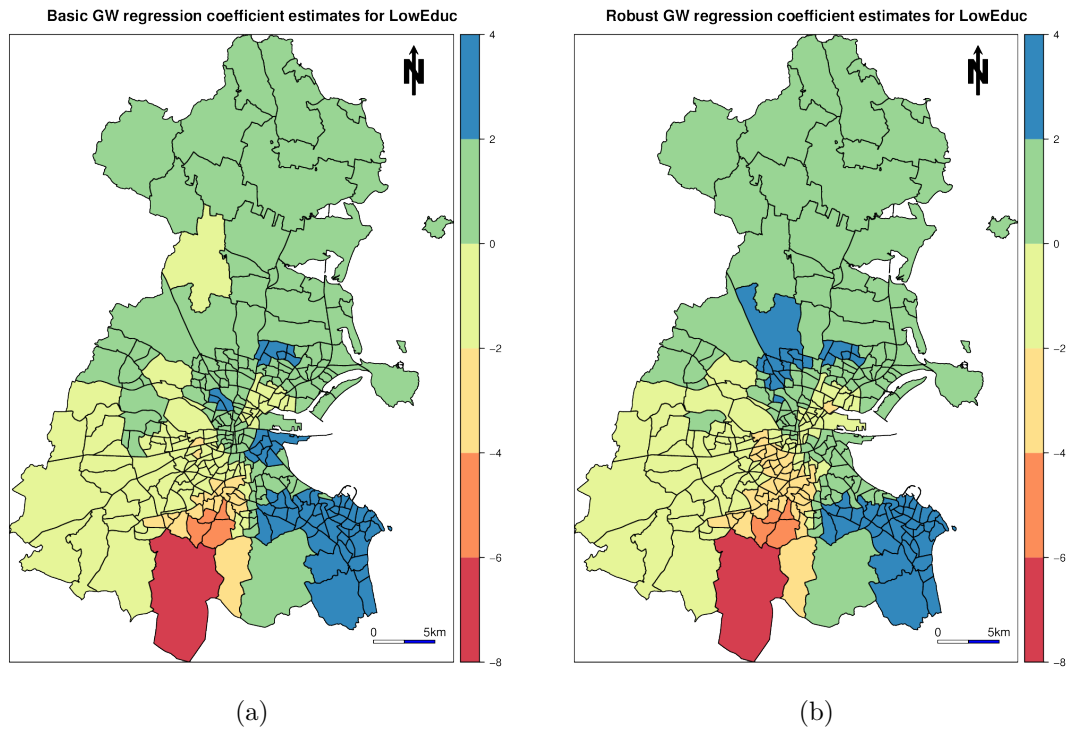


Figure 9: (a) Basic and (b) robust GW regression coefficient estimates for LowEduc.

variation, ranging from -7.67 to 3.41 . Its global regression coefficient estimate is -0.13 . Commands for a robust GW regression fit (the second, iterative approach) of the same model, using the same bandwidth, are also given. Here a slightly different set of coefficient estimates for LowEduc result (Figure 9b), to that found with the basic fit. Evidence for relationship non-stationarity is now slightly weaker, as the robustly estimated coefficients range from -7.74 to 2.57 , but the broad spatial pattern in these estimates remain largely the same.

```
R> names(gwr.res$SDF)
R> mypalette.6 <- brewer.pal(6, "Spectral")
R> spplot(gwr.res$SDF, "LowEduc", key.space = "right",
+   col.regions = mypalette.6, at = c(-8, -6, -4, -2, 0, 2, 4),
+   main = "Basic GW regression coefficient estimates for LowEduc",
+   sp.layout = map.layout)
R> rgwr.res <- gwr.robust(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter,
+   bw = bw.gwr.1, kernel = "bisquare", adaptive = TRUE, F123.test = TRUE)
R> print(rgwr.res)
R> spplot(rgwr.res$SDF, "LowEduc", key.space = "right",
+   col.regions = mypalette.6, at = c(-8, -6, -4, -2, 0, 2, 4),
+   main = "Robust GW regression coefficient estimates for LowEduc",
+   sp.layout = map.layout)
```


7. GW regression and addressing local collinearity

7.1. Introduction

A problem which has long been acknowledged in regression modelling is that of collinearity among the predictor (independent) variables. The effects of collinearity include a loss of precision and a loss of power in the coefficient estimates. Collinearity is potentially more of an issue in GW regression because: (i) its effects can be more pronounced with the smaller spatial samples used in each local estimation and (ii) if the data are spatially heterogeneous in terms of its correlation structure, some localities may exhibit collinearity while others may not. In both cases, collinearity may be a source of problems in GW regression even when no evidence is found for collinearity in the global model (Wheeler and Tiefelsdorf 2005; Wheeler 2007, 2013a). A further complication is that in the case of a predictor which has little local spatial variation, the possibility of collinearity with the intercept term is raised (Wheeler 2007, 2010, 2013a). Simulation studies have indicated that in the presence of collinearity, GW regression may find patterns in the coefficients where no patterns are actually present (Wheeler and Tiefelsdorf 2005; Páez, Farber, and Wheeler 2011).

To this extent, diagnostics to investigate the nature of collinearity in a GW regression analysis should always be conducted; this includes finding: (a) local correlations amongst pairs of predictors; (b) local variance inflation factors (VIFs) for each predictor; (c) local variance decomposition proportions (VDPs); and (d) local design (or cross-product) matrix condition numbers; all at the same spatial scale of each local regression of the GW regression model. Accordingly, the following rules of thumb can be taken to indicate likely local collinearity problems in the GW regression fit: (a) absolute local correlations greater than 0.8 for a given predictor variable pair; (b) VIFs greater than 10 for a given predictor; (c) VDPs greater than 0.5; and (d) condition numbers greater than 30. Such diagnostics and associated rules of thumb are directly taken from the global regression case (Belsley, Kuh, and Welsch 1980; O'Brien 2007) and have been proposed in a GW regression context through the works of Wheeler and Tiefelsdorf (2005); Wheeler (2007). All four diagnostics can be found and mapped using the function `gwr.collin.diagno` in **GWmodel** and a similar function exists in the **gwr** R package, see Section 9. Here it should be noted that local correlations and local VIFs cannot detect collinearity with the intercept; Wheeler (2010) provides a useful example of this. To this extent, the combined use of local VDPs and local condition numbers are superior diagnostics to local correlations and local VIFs for investigating collinearity (Wheeler 2007).

There are several possible actions in the light of discovering high levels of collinearity. These include: (1) doing nothing, (2) removing the offending predictors, (3) transforming the predictors to some orthogonal form or (4) using a different non-stationary regression model. The removal of a predictor is recommended if it clearly creates a global and local collinearity issue. However removing a predictor is not ideal, when only a local collinearity effect is present. Transforming the predictors should be done locally, at the same spatial scale of the GW regression (e.g., with a GW PCA), rather than globally (with PCA). However, such operations entail a loss of meaning in the regression outputs. For local collinearity issues, the fourth option is the best option and here Wheeler (2007, 2009) proposed significant modifications to the GW regression model, designed to cater for the adverse effects of collinearity. In particular, penalized regression models were transferred to a local form with the GW ridge

regression model (Wheeler 2007, 2009) and the GW lasso (Wheeler 2009). Both penalized GW models are biased estimators, providing no measures of coefficient uncertainty, but the GW lasso has the advantage in that it can also provide a local model selection function. In the presence of collinearity, both penalized GW models should provide more accurate local coefficient estimates, than that found with basic GW regression. Thus an investigation of relationship non-stationary should be more assured in this respect.

In this section, our aim is to present the use of local condition numbers as a diagnostic for local collinearity and also to relate this diagnostic to the ridge parameter of a GW ridge regression model. This relationship enables us to provide an alternative GW ridge regression to that demonstrated in Wheeler (2007). We call our new model, GW regression with a locally-compensated ridge term. This model differs to any existing GW ridge regression, in that: (A) it fits local ridge regressions with their own ridge parameters (i.e., the ridge parameter varies across space) and (B) it only fits such ridge regressions at locations where the local condition number is above a user-specified threshold. Thus a biased local estimation is not necessarily used everywhere; only at locations where collinearity is likely to be an issue. At all other locations, the usual un-biased estimator is used. **GWmodel** functions that we demonstrate are `bw.gwr.lcr`, to optimally estimate the bandwidth, and `gwr.lcr`, to fit the locally-compensated GW regression.

7.2. Ridge regression

A method to reduce the adverse effects of collinearity in the predictors of a linear model is ridge regression (Hoerl 1962; Hoerl and Kennard 1970). Other methods include principal components regression and partial least squares regression (Frank and Friedman 1993). In ridge regression the estimator is altered to include a small change to the values of the diagonal of the cross-product matrix. This is known as the ridge, indicated by λ in the following equation:

$$\hat{\beta} = \left(X^T X + \lambda I \right)^{-1} X^T Y$$

The effect of the ridge is to increase the difference between the diagonal elements of the matrix and the off-diagonal elements. As the off-diagonal elements represent the co-variation in the predictors, the effect of the collinearity among the predictors in the estimation is lessened. The price of this is that $\hat{\beta}$ becomes biased, and the standard errors (and associated t -values) of the estimates are no longer available. Of interest is the value to be given to the ridge parameter; Lee (1987) presents an algorithm to find a value which yields the best predictions.

7.3. GW regression with local compensation

There exists a link between the definition of the condition number for the cross-product matrix ($X^T X$) and the ridge parameter based on the observation that if the eigenvalues of $X^T X$ are $\epsilon_1, \epsilon_2, \dots, \epsilon_p$ then the eigenvalues of $X^T X + \lambda I$ are $\epsilon_1 + \lambda, \epsilon_2 + \lambda, \dots, \epsilon_p + \lambda$. The condition number κ of a square matrix is defined as ϵ_1/ϵ_p , so the condition number for the ridge-adjusted matrix will be $\epsilon_1 + \lambda/\epsilon_p + \lambda$. By re-arranging the terms, the ridge adjustment that will be required to yield a particular condition number κ is $\lambda = \{(\epsilon_1 - \epsilon_p)/(\kappa - 1)\} - \epsilon_p$. Thus given the eigenvalues of the un-adjusted matrix, and the desired condition number, we can determine the value of the ridge which is required to yield that condition number.

For GW regression, this can be applied to the GW cross-product matrix, which permits a

local compensation of each local regression model, so that the local condition number never exceeds a specified value of κ . The condition numbers for the un-adjusted matrices may also be mapped to give an indication of where the analyst should take care in interpreting the results, or the local ridge parameters may also be mapped. For cases where collinearity is as much an issue in the global regression as in the GW regression; the local estimations will indicate precisely where the collinearity is a problem. The estimator for this locally compensated ridge (LCR) GW regression model is:

$$\hat{\beta}(u_i, v_i) = \left(X^\top W(u_i, v_i) X + \lambda I(u_i, v_i) \right)^{-1} X^\top W(u_i, v_i) Y$$

where $\lambda I(u_i, v_i)$ is the locally-compensated value of λ at location (u_i, v_i) . Observe that the same approach to estimating the bandwidth in the basic GW regression (Section 6) can be applied to the LCR GW regression model. For a cross-validation approach, the bandwidth is optimized to yield the best predictions. Collinearity tends to have a greater affect on the coefficient estimates rather than the predictions from the model, so in general, little is lost when using the locally-compensated form of the model. Details on this and an alternative locally-compensated GW regression can be found in [Brunsdon, Charlton, and Harris \(2012\)](#), where both models are performance tested within a simulation experiment.

7.4. Example

We examine the use of our local compensation approach with the same GW regression that is specified in Section 6, where voter turnout is a function of the eight predictor variables of the Dublin election data. For the corresponding global regression, the `vif` function in the `car` package ([Fox and Weisberg 2011](#)) computes VIFs using the method outlined in [Fox and Monette \(1992\)](#). These global VIFs are given below and (noting their drawbacks given above) suggest that weak collinearity exists within this data.

```
R> library("car")
R> lm.global <- lm(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter)
R> summary(lm.global)
R> vif(lm.global)
```

DiffAdd	LARent	SC1	Unempl	LowEduc	Age18_24	Age25_44	Age45_64
3.170044	2.167172	2.161348	2.804576	1.113033	1.259760	2.879022	2.434470

In addition, the PCA from Section 5 suggests collinearity between `DiffAdd`, `LARent`, `Unempl`, `Age25_44`, and `Age45_64`. As the first component accounts for some 36% of the variance in the data set, and of those components with eigenvalues greater than 1, the proportion of variance accounted for is 73.6%, we might consider removing variables with higher loadings. However for the purposes of illustration, we decide to keep the model as it is. Further global findings are of note, in that the correlation of turnout with `Age45_64` is positive, but the sign of the global regression coefficient is negative. Furthermore, only four of the global regression predictors are significant. Unexpected sign changes and relatively few significant variables are both indications of collinearity.

We can measure the condition number of the design matrix using the method outlined in [Belsley et al. \(1980\)](#). The method, termed BKW, requires that the columns of the matrix

are scaled to have length 1; the condition number is the ratio of the largest to the smallest singular value of this matrix. The following code implements the BKW computations, where X is the design matrix consisting of the predictor variables and a column of 1s.

```
R> X <- as.matrix(cbind(1, Dub.voter@data[, 4:11]))
R> BKWcn <- function(X) {
+   p <- dim(X)[2]
+   Xscale <- sweep(X, 2, sqrt(colSums(X^2)), "/")
+   Xsvd <- svd(Xscale)$d
+   Xsvd[1] / Xsvd[p]
+ }
R> BKWcn(X)
```

```
[1] 41.06816
```

The BKW condition number is found to be 41.07 which is high, indicating that collinearity is at least, a global problem for this data. We can experiment by removing columns from the design matrix and test which variables appear to be the source of the collinearity. For example, entering:

```
R> BKWcn(X[, c(-2, -8)])
```

```
[1] 18.69237
```

allows us to examine the effects of removing both `DiffAdd` and `Age25_44` as sources of collinearity. The reduction of the BKW condition number to 18.69 suggests that removing these two variables is a useful start. However, for demonstration purposes, we will persevere with the collinear (full specification) model, and now re-examine its GW regression fit, the one already fitted in Section 6. The main function to perform this collinearity assessment is `gwr.lcr`, where we aim to compare the coefficient estimates for the un-adjusted basic GW regression with those from a LCR GW regression.

In the first instance, we can use this function to find the global condition number (as that found with the global regression). This can be done simply by specifying a box-car kernel with a bandwidth equal to the sample size. This is equivalent to fitting n global models. Inspection of the results from the spatial data frame show that the condition numbers are all equal to 41.07, as hoped for. The same condition number is outputted by the **ArcGIS** Geographically weighted Regression tool in the Spatial Statistics Toolbox (ESRI 2013). Commands to conduct this check on the behavior of the `lcr.gwr` function are as follows:

```
R> nobs <- dim(Dub.voter)[1]
R> lcrm1 <- gwr.lcr(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl + LowEduc +
+   Age18_24 + Age25_44 + Age45_64, data = Dub.voter, bw = nobs,
+   kernel = "boxcar", adaptive = TRUE)
R> summary(lcrm1$SDF$Local_CN)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
41.07	41.07	41.07	41.07	41.07	41.07

To obtain local condition numbers for a basic GW regression without a local compensation, we use the `bw.gwr.lcr` function to optimally estimate the bandwidth, and then `gwr.lcr` to estimate the local regression coefficients and the local condition numbers⁴. To match that of Section 6, we specify an adaptive bi-square kernel. Observe that the bandwidth for this model can be exactly the same as that obtained using `bw.gwr`, the basic bandwidth function. With no local compensation (i.e., local ridges of zero), the cross-products matrices will be identical, but only provided the same optimization approach is specified. Here we specify a cross-validation (CV) approach, as the AICc approach is currently not an option in the `bw.gwr.lcr` function. Coincidentally, for our basic GW regression of Section 6, a bandwidth of $N = 109$ results for both CV and AICc approaches. Commands to output the local condition numbers from our basic GW regression, and associated model comparisons are as follows:

```
R> lcrm2.bw <- bw.gwr.lcr(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter,
+   kernel = "bisquare", adaptive = TRUE)
R> lcrm2.bw

[1] 109

R> lcrm2 <- gwr.lcr(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl + LowEduc +
+   Age18_24 + Age25_44 + Age45_64, data = Dub.voter, bw = lcrm2.bw,
+   kernel = "bisquare", adaptive = TRUE)
R> summary(lcrm2$SDF$Local_CN)

    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 32.88  52.75   59.47   59.28  64.85  107.50

R> gwr.cv.bw <- bw.gwr(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter,
+   approach = "CV", kernel = "bisquare", adaptive = TRUE)
R> gwr.cv.bw

[1] 109

R> mypalette.7 <- brewer.pal(8, "Reds")
R> spplot(lcrm2$SDF, "Local_CN", key.space = "right",
+   col.regions = mypalette.7, at = seq(30, 110, length = 9),
+   main = "Local condition numbers from basic GW regression",
+   sp.layout = map.layout)
```

Thus the local condition numbers can range from 32.88 to 107.50, all worryingly large everywhere. Whilst the local estimations are potentially more susceptible to collinearity than the global model, we might consider removing some of the variables which cause problems globally. The maps will show where the problem is worst, and where action should be concentrated. The local condition numbers for this estimation are shown in Figure 10a.

⁴Local condition numbers can also be found using the function `gwr.collin.diagno`.

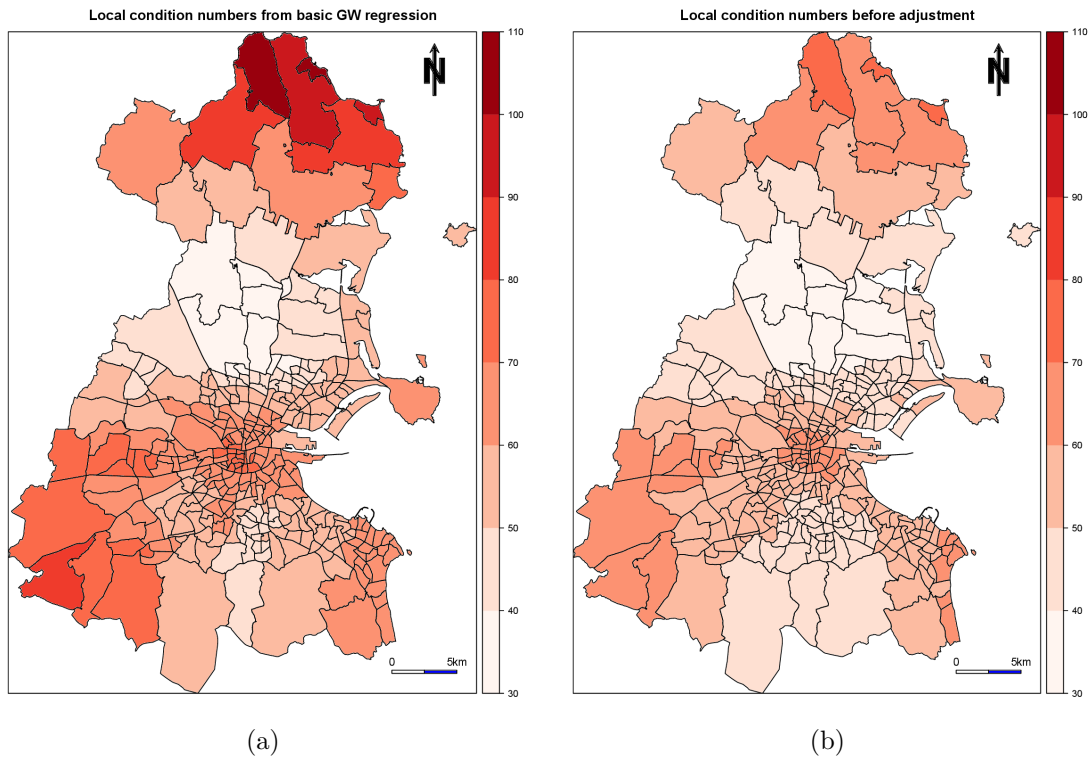


Figure 10: Local condition numbers from: (a) basic GW regression and (b) before adjustment.

We can now calibrate a LCR GW regression, where the application of local ridge adjustment to each local $X^T W(u_i, v_i) X$ matrix, only occurs at locations where the local condition number exceeds some user-specified threshold. At these locations, the local compensation forces the condition numbers not to exceed the same specified threshold. In this case, we follow convention and specify a threshold of 30. As a consequence, a local ridge term will be found at all locations, for this particular example. The `lambda.adjust = TRUE` and `cn.thresh = 30` arguments in the `gwr.lcr` function are used to invoke the local compensation process, as can be seen in following commands:

```
R> lcrm3.bw <- bw.gwr.lcr(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter,
+   kernel = "bisquare", adaptive = TRUE, lambda.adjust = TRUE,
+   cn.thresh = 30)
R> lcrm3.bw

[1] 157

R> lcrm3 <- gwr.lcr(GenEl2004 ~ DiffAdd + LARent + SC1+ Unempl + LowEduc +
+   Age18_24 + Age25_44 + Age45_64, data = Dub.voter, bw = lcrm3.bw,
+   kernel = "bisquare", adaptive = TRUE, lambda.adjust = TRUE,
+   cn.thresh = 30)
R> summary(lcrm3$SDF$Local_CN)
```


Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
34.34	47.08	53.84	52.81	58.66	73.72

```
R> summary(lcrm3$SDF$Local_Lambda)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.01108	0.03284	0.04038	0.03859	0.04506	0.05374

```
R> spplot(lcrm3$SDF, "Local_CN", key.space = "right",
+   col.regions = mypalette.7, at = seq(30, 110, length = 9),
+   main = "Local condition numbers before adjustment",
+   sp.layout = map.layout)
R> spplot(lcrm3$SDF, "Local_Lambda", key.space = "right",
+   col.regions = mypalette.7, cuts = 7, sp.layout = map.layout,
+   main = "Local ridge terms for LCR GW regression")
```

Observe that the bandwidth for the locally compensated GW regression is larger at $N = 157$, than for the un-adjusted (basic) GW regression (at $N = 109$). We could have specified the bandwidth from the un-adjusted model, but this would not provide the best fit. The larger bandwidth provides greater smoothing. Observe also that the local condition numbers (the `Local_CN` outputs) from this model are the local condition numbers *before* the adjustment (or compensation)⁵. They will tend to be smaller than those for the basic model because we are using a larger bandwidth. They will tend to that of the global model (i.e., 41.07).

The `Local_Lambda` outputs are the local ridge estimates used to adjust the local cross-products matrices. Both the local condition numbers and the local ridges can be mapped to show where the GW regression has applied different levels of adjustment in relation to the different levels of collinearity among the predictors. The local condition numbers are mapped in Figure 10b, and the local ridges in Figure 11a for our LCR GW regression. The greatest adjustments were required in central Dublin, and in the north, south-west and south-east extremities of the study area.

Figure 11b plots the adjusted coefficient estimates for `LARent` from the locally compensated model, against those from the corresponding basic model. The general pattern would appear to be that the larger coefficients for the basic model are reduced in magnitude, and that the smaller coefficients are raised. The relationship is non-linear and a loess fit is shown in the plot. Commands for this comparison are as follows:

```
R> gwr.cv <- gwr.basic(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64, data = Dub.voter
+   bw = gwr.cv.bw, kernel = "bisquare", adaptive = TRUE)
R> small <- min(min(gwr.cv$SDF$LARent), min(lcrm3$SDF$LARent))
R> large <- max(max(gwr.cv$SDF$LARent), max(lcrm3$SDF$LARent))
R> plot(gwr.cv$SDF$LARent, lcrm3$SDF$LARent,
+   main = "LARent coefficients: basic vs. locally compensated",
+   xlab = "GW regression coefficient",
+   ylab = "LCR GW regression coefficient",
+   xlim = c(small, large), ylim = c(small, large))
```

⁵After the adjustment, the local condition numbers will all equal 30 in this case.

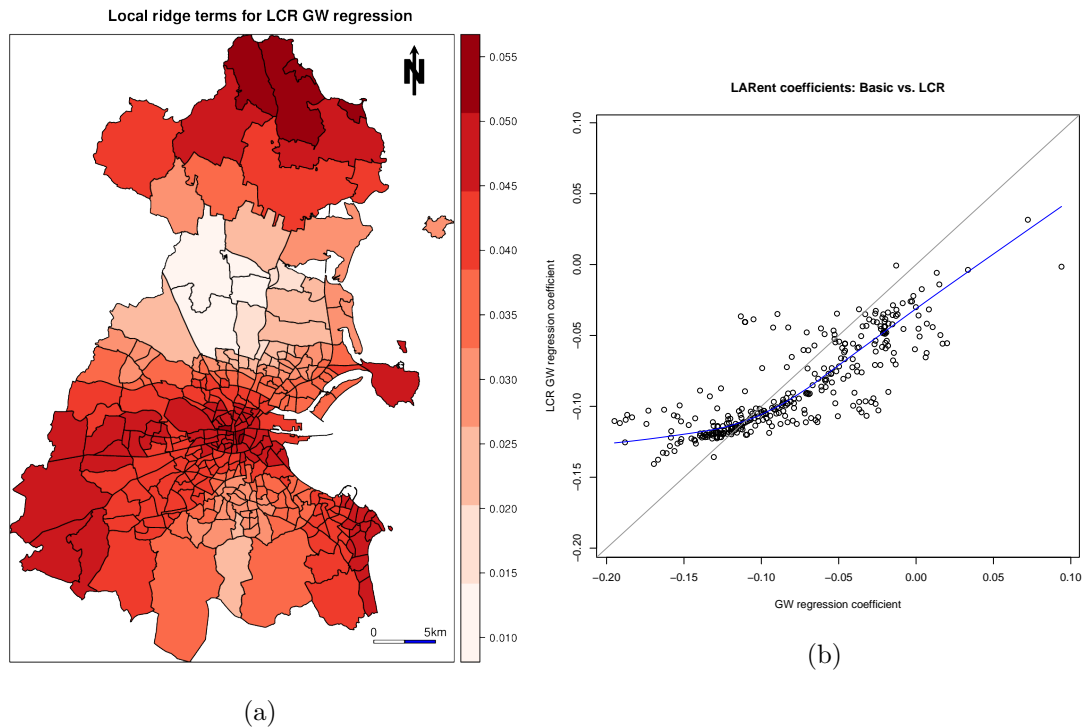


Figure 11: (a) Local ridge terms and (b) comparison of coefficient estimates for `LARent`.

```
R> lines(lowess(gwr.cv$SDF$LARent, lcrm3$SDF$LARent), col = "blue")
R> abline(0, 1, col = "gray60")
```

Model building with collinear data

If we explore the local condition numbers for models with different structures, it may be possible to build GW regression models which avoid collinearity. Here, we code a function to calibrate and then estimate a basic (un-adjusted) GW regression. This function can then be used to assess various forms of the model, where the output each time is a vector of local condition numbers for the model that has been fitted. This function is presented as follows, together with an example model run.

```
R> test.CN <- function(model, data) {
+   lcrmx.bw <- bw.gwr.lcr(model, data = data, kernel = "bisquare",
+     adaptive = TRUE)
+   print(model)
+   print(lcrmx.bw)
+   lcrmx <- gwr.lcr(model, data = data, bw = lcrmx.bw,
+     kernel = "bisquare", adaptive = TRUE)
+   print(summary(lcrmx$SDF$Local_CN))
+   lcrmx$SDF$Local_CN
+ }
R> data <- Dub.voter
```

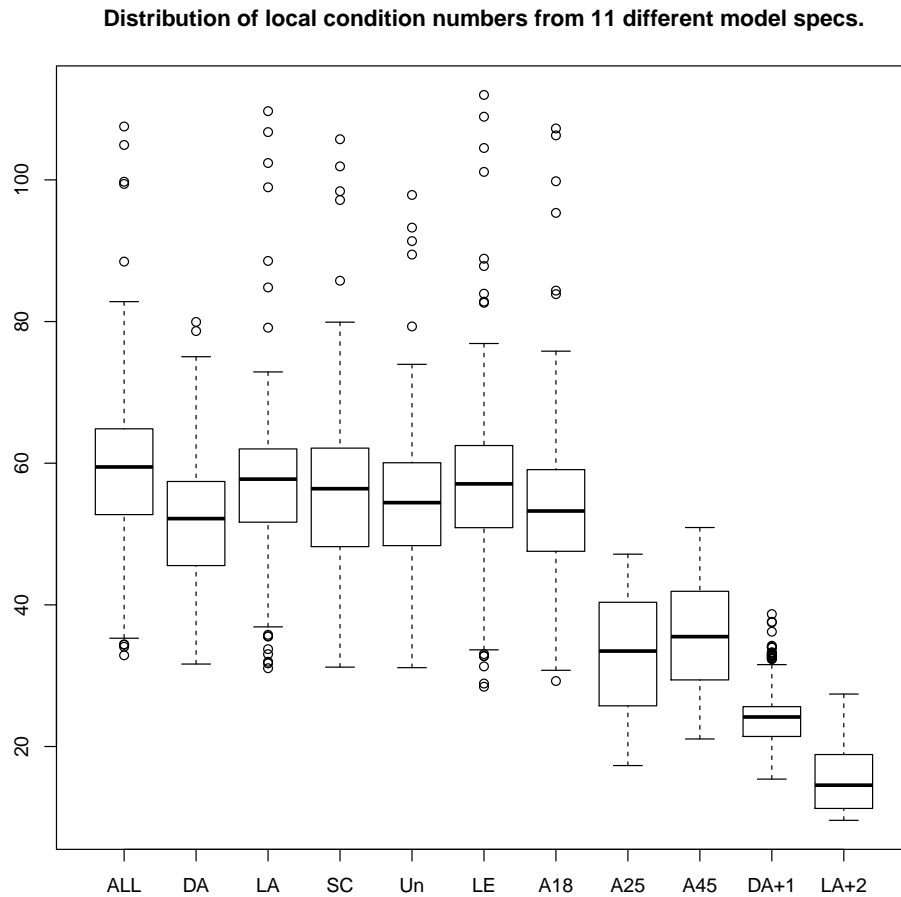
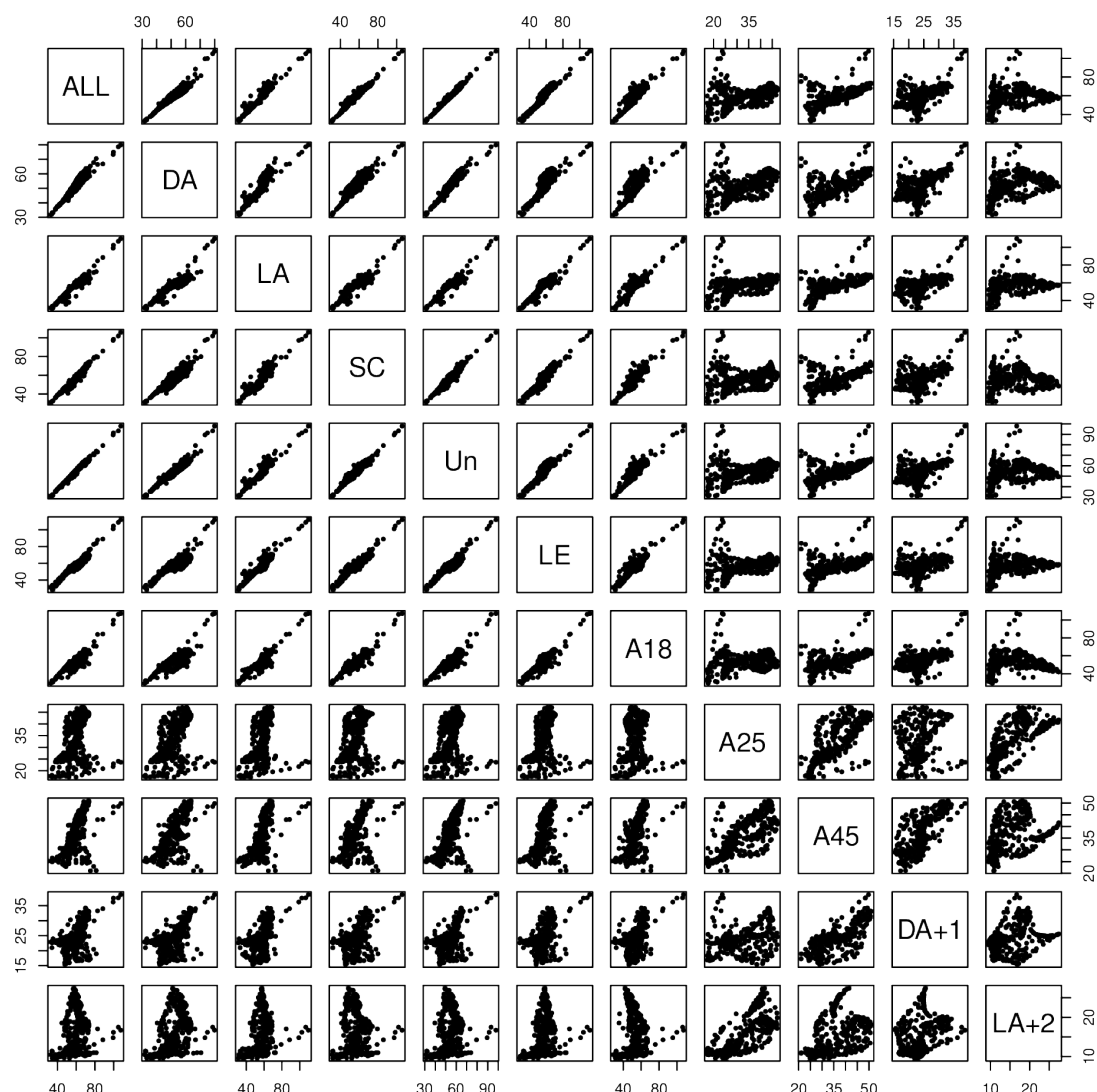



Figure 12: Distribution of local condition numbers from 11 different GW regression fits.

```
R> model <- as.formula(GenEl2004 ~ DiffAdd + LARent + SC1 + Unempl +
+   LowEduc + Age18_24 + Age25_44 + Age45_64)
R> AllD <- test.CN(model, data)
```

On using this function with eleven different GW regression models, Figure 12 shows the boxplots of the local condition numbers from GW regressions with: (i) all variables (ALL), (ii) removing each variable in turn (DiffAdd, LARent, SC1, Unempl, LowEduc, Age18_24, Age25_44, Age45_64), (iii) removing DiffAdd and Age45_64 together, and (iv) removing LARent, Age25_44 and Age45_64 together. The last grouping was suggested by the output of the global PCA from Section 5. Removing variables individually has little effect on the local condition number distributions, although removing the last two age variables induces a noticeable drop. Removing the most collinear variables produces a model where no local condition number is above 30.

Figure 13 shows the local condition number distributions as a scatterplot matrix. Here the least effective variable removals have high correlations between the condition number distributions, whereas removing the most collinear variables tends to provide lower condition number correlations with other model forms. This opens up the possibility of semi-automating the model building process to yield a GW regression with acceptably low levels of collinearity.



7.5. LCR GW regression vs. previous penalized GW regression models

It is important to clarify the difference between our LCR GW regression (say, LCR-GWR) and the GW ridge regression (GWRR) demonstrated in [Wheeler \(2007\)](#). Essentially, LCR-GWR is more locally-focused than GWRR. GWRR similarly applies a local compensation, but for each local regression, the ridge parameter is global (i.e., it does not vary across space). This global ridge is used everywhere, ensuring that all local regressions of GWRR are biased. Whereas for LCR-GWR, local ridge parameters are used, and they are only used at locations where they are most needed (as set by a condition number threshold). At other locations, ridges of zero are specified. Thus depending on the condition number threshold set, not all of the local regressions of a LCR-GWR model are necessarily biased. Wheeler actually noted this

short-coming of GWRR and suggested the use of local ridge terms (Wheeler 2007, p.2480), but provided no details on such a model's implementation. Thus the advance for LCR-GWR is the actual implementation of a such a model, and in particular, the relating of the local condition number to the local ridge parameter (Section 7.3).

LCR-GWR should not only be placed in context with GWRR, but also the GW lasso (GWL). Ridge regression and the lasso (Tibshirani 1996) both penalize or shrink regression coefficients, so as to reduce collinearity effects, where the lasso shrinks the least significant coefficients to zero, and in doing so, additionally (and simultaneously) provides a model selection function. Wheeler (2009) provides two versions of GWL; one that estimates a single (global) lasso parameter to control coefficient shrinkage (GWL-global), and one that estimates multiple (local) lasso parameters (GWL-local). Thus GWL-global can be viewed as an alternative to GWRR (as both are globally-focused), whilst GWL-local can be viewed as an alternative to LCR-GWR (as both are locally-focused).

A simple way to demonstrate the differences between the competing models is to run them (together with a basic GW regression), using the same data and the same kernel weighting specification. Here the functions, `gwrr.est` and `gwl.est` from the `gwrr` R package are needed to fit the GWRR and GWL models, respectively. Correspondence with the author of `gwrr` indicated that `gwl.est` only fits GWL-local models, and that an option for GWL-global is not provided. Furthermore, `gwrr.est`, `gwl.est` and `gwr.est` (a basic GW regression function in `gwrr`), only allow for fixed bandwidths with Gaussian or exponential kernel functions (see Section 9). As the bandwidth for GWL-local is optimally estimated (via cross-validation) within the `gwl.est` function, and cannot be user-specified, we use this particular bandwidth for all models. Thus, only GWL-local will be correctly specified with respect to its weighting function. For a data set, we choose the `columbus` neighborhood crime rate data available in `gwrr`.

Commands to conduct this rudimentary comparison are given below, where an exponential kernel bandwidth of 1.68 is used for all models. For GWRR and LCR-GWR, only the latter provides local ridges, 16 of which are non-zero (as their corresponding local condition numbers are above 30). The only outputs that are directly comparable between all four models, are the local regression coefficients themselves. For example, if we view the coefficients for `income`, 26 are zero with GWL-local, indicating that this variable is not needed in the corresponding local regressions. If we compare the `income` coefficients for the penalized models with those from the basic model, both GWRR and GWL-local apply some form of penalty at all locations, whereas LCR-GWR only penalizes at certain (in this case, 16) locations. Thus as would be expected, coefficients from the basic GW regression have the strongest correlation with those from the LCR-GWR model. An objective and detailed comparison of all such penalized models, using simulated data, is a subject of our current research (Brunsdon *et al.* 2012).

```
R> library("gwrr")
R> data("columbus")
R> locs <- cbind(columbus$x, columbus$y)
R> columbus.spdf <- SpatialPointsDataFrame(locs, columbus)
R> gwl.comp <- gwl.est(crime ~ income + houseval, locs, columbus, "exp")
R> gwl.comp$phi
```

```
[1] 1.678067
```

```
R> summary(gwl.comp$beta[2, ])

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-2.6690 -1.1400  0.0000 -0.5731  0.0000  0.6526

R> lcr.gwr.comp <- gwr.lcr(crime ~ income + houseval, data = columbus.spdf,
+   bw = gwl.comp$phi, kernel = "exponential", lambda.adjust = TRUE,
+   cn.thresh = 30)
R> summary(lcr.gwr.comp$SDF$Local_CN)

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 8.245  18.150  22.000  32.120  39.800 109.000

R> summary(lcr.gwr.comp$SDF$Local_Lambda)

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.000000 0.000000 0.000000 0.008032 0.014580 0.042930

R> gwrr.comp <- gwrr.est(crime ~ income + houseval, locs, columbus,
+   "exp", bw = gwl.comp$phi)
R> gwrr.comp$lambda

[1] TRUE

R> gwr.comp <- gwr.est(crime ~ income + houseval, locs, columbus,
+   "exp", bw = gwl.comp$phi)
R> cor(cbind(gwl.comp$beta[2, ], lcr.gwr.comp$SDF$income,
+   gwrr.comp$beta[2, ], gwr.comp$beta[2, ]))

      [,1]      [,2]      [,3]      [,4]
[1,] 1.0000000 0.5923885 0.2228593 0.6160248
[2,] 0.5923885 1.0000000 0.2165015 0.9187365
[3,] 0.2228593 0.2165015 1.0000000 0.3579617
[4,] 0.6160248 0.9187365 0.3579617 1.0000000
```

7.6. Concluding remarks: Criticisms and Bayesian models

Collinearity is a problem in any form of regression modelling and the importance of assessing and taking action has been raised by the many commentaries on local collinearity in GW regression (Wheeler and Tiefelsdorf 2005; Wheeler 2007, 2009, 2010, 2013a; Griffith 2008; Páez *et al.* 2011). Collinear data require care and the tools available in **GWmodel** should help the analyst assess the magnitude of the problem, where it is a problem, and lead the analyst to take appropriate action.

Properly accounting for the effects of collinearity in GW regression is one of three recommendations for the practical application of GW regression, given in the simulation study of Páez

et al. (2011). A second and similarly important recommendation is that the bandwidth should be chosen with care. In our view, if an optimally found bandwidth tends to its maximum, then the underlying process should be viewed as stationary. Optimal bandwidths that are very small should also be viewed suspiciously, suggesting a underlying random process. In this respect, **GWmodel** provides additional functions to more fully interrogate bandwidth choice, complementing the automated procedures (see Section 10). It should always be noted that an optimal bandwidth for GW regression is based on the accurate prediction of the response variable, not (as it ideally should be) the accurate prediction of the coefficients (which after-all, are the ones used to assess relationship non-stationarity). A further consideration is that bandwidth size will directly affect the nature of local collinearity. In this respect, *Brunsdon et al.* (2012) propose an alternative locally-compensated GW regression, where the bandwidth is increased only at locations most seriously effected by collinearity. In essence, the bandwidth selection function in any GW model should always be thoroughly investigated, and can be considered analogous to a thorough investigation of the variogram in geostatistics; where both investigations aim to identify spatial structure in some way (see also *Cressie* 1989).

A third recommendation of *Páez et al.* (2011) concerns sample size, where it is not recommended to use GW regression for data sets of $n < 160$. In our view, this threshold could be set lower, but provided a thorough and thoughtful GW regression analysis is undertaken, using many of the tools described here. As indicated by *Páez et al.* (2011), the interpretation of the results should "proceed with extreme caution", in such cases. Observe that this third recommendation can be just as applicable to any non-stationary spatial model, not just GW regression. Considering all three recommendations together, ensures that GW regression should never be viewed as a 'black box' technique.

Once a thorough GW regression analysis has been conducted, the exploratory and hopefully, reliable results may need to be extended for statistical inference. Here, GW regression does not provide a coherent inferential framework, since no one model exists. Instead GW regression yields a collection of models, calibrated with the same data, but with different weighting schemes. In this respect, any hypothesis testing with GW regression (for example an investigation of the local (pseudo) t -values) should only ever be viewed as explorative.

To this extent, it may be worthwhile to consider a Bayesian spatially varying coefficient (SVC) model (*Gelfand, Kim, Sirmans, and Banerjee* 2003; *Assuncao* 2003; *Wheeler and Calder* 2007; *Waller, Zhu, Gotway, Gorman, and Grunewald* 2007; *Wheeler and Waller* 2009; *Finley* 2011). These non-stationary regressions do provide a coherent inferential framework (as a single model exists) and re-assuringly, do not suffer from the local collinearity problems found in GW regression. Bayesian SVC models do however suffer from their computational complexity, where often only the simplest models can be run (e.g., with only two or three predictors). Furthermore, software for their implementation is often not readily available (*Finley* 2011, p.153). Until these modelling issues are resolved, it is likely that GW regression will continue to provide an important exploratory technique for the investigation of relationship non-stationarity; which on occasion, may help direct and parameterize, a subsequent Bayesian SVC model fit and analysis.

8. GW regression for spatial prediction

The use of GW regression as a spatial predictor has attracted much attention (e.g., the empirical evaluations of *Gao, Asami, and Chung* 2006; *Páez et al.* 2008; *Lloyd* 2010b), where

it has often performed well, relative to a geostatistical (kriging) alternative. More objective, simulation experiments can be found in [Harris, Fotheringham, Crespo, and Charlton \(2010b\)](#), where kriging performed the best, but GW regression still performed with merit. Conceptual links between GW regression and kriging are discussed in [Harris, Brunsdon, and Fotheringham \(2011b\)](#) and various hybrid predictors can be found in [Lloyd \(2010b\)](#); [Harris *et al.* \(2010b\)](#); [Harris and Juggins \(2011\)](#), where GW regression can provide a useful trend component of a kriging model.

Thus studies have demonstrated value in using GW regression as a spatial predictor. This is not surprising given that GW regression and associated tools, directly stem from the non-parametric, locally weighted regression models of [Cleveland \(1979\)](#); [Cleveland and Devlin \(1988\)](#), for curve fitting and interpolation. Furthermore, when used as spatial predictor, the many criticisms levelled at GW regression, such as problems with collinearity (Section 7), are not expected to be of such importance. In this respect, our current research is investigating this topic, via simulated data experiments.

To use GW regression as a spatial predictor is straight-forward, a GW regression prediction at a location s can be found using:

$$\hat{y}_{GW}(s) = x(s)^\top \hat{\beta}(s)$$

where $x(s)$ and $\hat{\beta}(s)$ are each vectors of independent data values and parameter estimates, respectively. Following [Leung *et al.* \(2000\)](#), the corresponding GW regression prediction variance at s can be estimated using:

$$\sigma_{GW}^2(s) = \text{VAR} \{ \hat{y}(s) - y(s) \} = \hat{\sigma}^2 [1 + S(s)]$$

where

$$S(s) = x(s)^\top \left[X^\top W(s) X \right]^{-1} X^\top W^2(s) X \left[X^\top W(s) X \right]^{-1} x(s)$$

Here an unbiased estimate of the residual variance is taken as $\hat{\sigma} = \text{RSS}/(n - \text{ENP})$, where RSS is the residual sum of squares and ENP is the effective number of parameters of the GW regression model. Observe that the prediction variance for GW regression is for a single observation and not for the mean, and in doing so, is directly comparable with that found with kriging, in say, the **gstat** R package ([Pebesma 2004](#)).

8.1. Example

To demonstrate GW regression as spatial predictor, we use the **EWHP** data set. Here our aim is to predict the dependent variable, house price (**PurPrice**) using a subset of the nine independent variables described in Section 2, each of which reflect some hedonic characteristic of the property. A correlation analysis reveals that **FlrArea** (floor area or the effective size of the property) provides the strongest correlation with house price at $\rho = 0.65$. For demonstration purposes, we choose to focus our prediction models using only this single hedonic variable. Two additional R package are required for this analysis: **ModelMap** ([Freeman and Frescino 2009](#)) and **gstat** ([Pebesma 2004](#)).

```
R> library("ModelMap")
R> library("gstat")
```

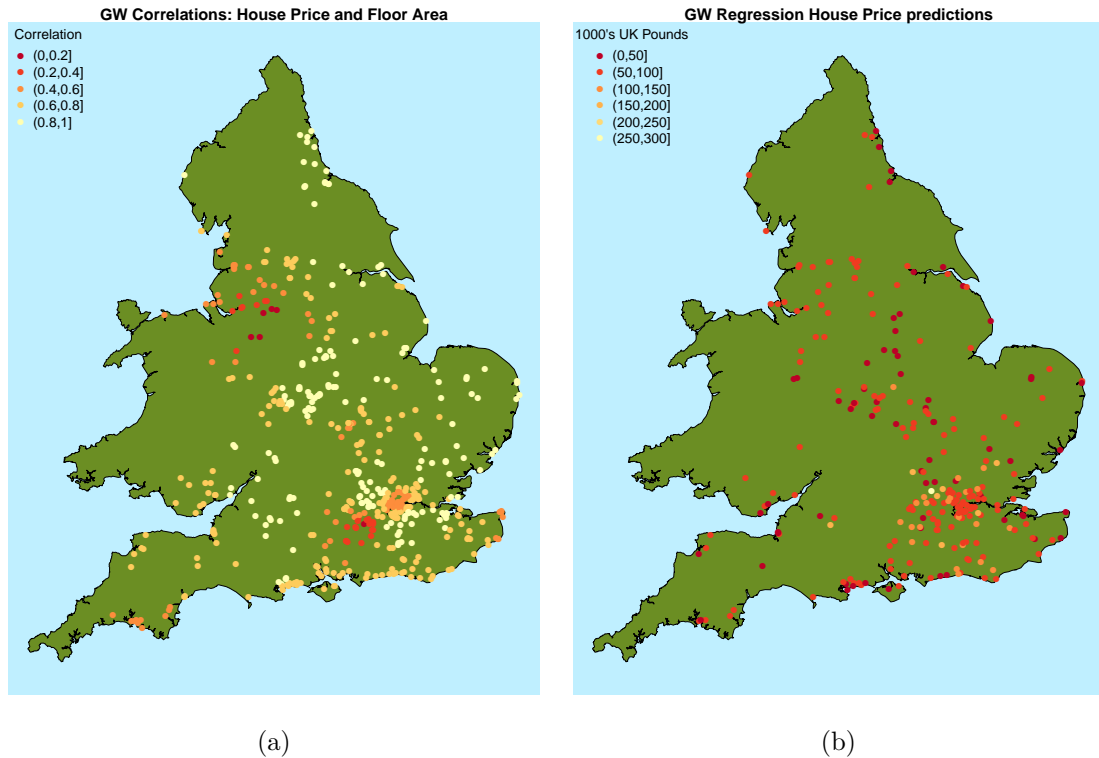


Figure 14: (a) GW correlations between house price and floor area and (b) GW regression predictions of house price.

A GW correlation analysis between house price and our retained hedonic variable⁶ can be conducted using the `gwss` function, where GW correlations (and other GW summary statistics) are specified using an adaptive bi-square kernel with a bandwidth of $N = 52$ (approximately 10% of the sample data). The `quick.map` function allows the GW correlations to be mapped. Figure 14a displays the resultant map, where the relationship between house price and floor area tends to weaken in rural areas. This is expected, as large properties are more likely to sell for a premium in urban areas (with the notable exception of some properties located in central London). Commands for this analysis are as follows:

```
R> ewhp.spdf <- SpatialPointsDataFrame(ewhp[, 1:2], ewhp)
R> data("EWOutline")
R> gw.sum.stats <- gwss(ewhp.spdf, vars = c("PurPrice", "FlrArea"),
+   kernel = "bisquare", adaptive = TRUE, bw = 52)
R> quick.map <- function(spdf, var, legend.title, main.title) {
+   x <- spdf@data[,var]
+   cut.vals <- pretty(x)
+   x.cut <- cut(x, cut.vals)
+   cut.levels <- levels(x.cut)
+   cut.band <- match(x.cut, cut.levels)
```

⁶Observe that discarding hedonic variables that correlate weakly at the global scale does not directly entail similarly weak correlations, locally. As such, a comprehensive analysis would have locally investigated all relationships with their respective GW correlations (see Section 4).


```

+   colors <- rev(brewer.pal(length(cut.levels), "YlOrRd"))
+   par(mar = c(1, 1, 1, 1))
+   plot(ewoutline, col = "olivedrab", bg = "lightblue1")
+   title(main.title)
+   plot(spdf, add = TRUE, col = colors[cut.band], pch = 16)
+   legend("topleft", cut.levels, col = colors, pch = 16, bty = "n",
+         title = legend.title)
+ }
R> quick.map(gw.sum.stats$SDF, "Corr_PurPrice.FlrArea",
+   "Correlation", "GW Correlations: House Price and Floor Area")

```

GW correlations provide evidence of non-stationarity in a specific house price relationship and as such, the use of GW regression to predict house price is worth pursuing. To this extent, we compare the prediction accuracy of a GW regression fit with that of its corresponding (ordinary least squares) global regression fit⁷. Here we split the EWHP data into model calibration and model validation data sets of equal size using the function `get.test` in the **ModelMap** R package. To find an optimal bandwidth for the GW regression, the function `bw.gwr` is used with the calibration data and an optimal adaptive bandwidth of $N = 34$ results (for a bi-square kernel, via the CV approach). We then parameterize the function `gwr.predict` with this bandwidth, to find the GW regression predictions and prediction variances at the validation sites. The necessary distance matrices are found using the function `gw.dist`. The corresponding global regression is also found using `gwr.predict`; and as a check for consistency, similarly found using functions `gstat` and `predict` from the **gstat** package. The commands used are as follows:

```

R> write.table(ewhp, "ewhp.csv", col.names = TRUE, row.names = FALSE,
+   sep = ",")
R> get.test(proportion.test = 0.5, "ewhp.csv", seed = 42, folder = getwd(),
+   qdata.trainfn = "ewhp_calib.csv", qdata.testfn = "ewhp_valid.csv")
R> ewhp_calib <- read.table("ewhp_calib.csv", header = TRUE, sep = ",")
R> attach(ewhp_calib)
R> ewhp_calib.spdf <- SpatialPointsDataFrame(ewhp_calib[, 1:2],
+   as.data.frame(ewhp_calib[c(3, 12)]))
R> dm.calib <- gw.dist(dp.locat = coordinates(ewhp_calib.spdf))
R> gwr.bw.cv <- bw.gwr(PurPrice ~ FlrArea, data = ewhp_calib.spdf,
+   approach = "CV", kernel = "bisquare", adaptive = TRUE, dMat = dm.calib)
R> ewhp_valid <- read.table("ewhp_valid.csv", header = TRUE, sep = ",")
R> attach(ewhp_valid)
R> ewhp_valid.spdf <- SpatialPointsDataFrame(ewhp_valid[, 1:2],
+   as.data.frame(ewhp_valid[c(3, 12)]))
R> dm.valid <- gw.dist(dp.locat = coordinates(ewhp_calib.spdf),
+   rp.locat = coordinates(ewhp_valid.spdf))
R> gwr.pred <- gwr.predict(PurPrice ~ FlrArea, data = ewhp_calib.spdf,
+   predictdata = ewhp_valid.spdf, bw = gwr.bw.cv, kernel = "bisquare",
+   adaptive = TRUE, dMat1 = dm.valid, dMat2 = dm.calib)

```

⁷A more complete analysis would also calibrate geostatistical/autocorrelation-based predictors for comparison (e.g., Páez *et al.* 2008, with respect to house price prediction).

Model	RMSPE	MAPE	Mean.ZS	SD.ZS
GW regression	27.03	17.55	-0.04	1.10
Global regression (GWmodel)	31.58	20.65	0.16	1.10
Global regression (gstat)	31.58	20.65	0.16	1.10

Table 2: Performance results for GW Regression as a spatial predictor (RMSPE and MAPE in £1000).

```
R> ols.pred.gwmodel <- gwr.predict(PurPrice ~ FlrArea,
+   data = ewhp_calib.spdf, predictdata = ewhp_valid.spdf, bw = 519,
+   kernel = "boxcar", adaptive = TRUE, dMat1 = dm.valid, dMat2 = dm.calib)
R> ols <- gstat(id = "mlr", formula = PurPrice ~ FlrArea,
+   loc = ~ Easting + Northing, data = ewhp_calib)
R> ols.pred.gstat <- predict(ols, newdata = ewhp_valid, BLUE = TRUE)
```

Performance results are reported in Table 2, in terms of prediction accuracy and prediction uncertainty accuracy. Prediction accuracy is measured by the root mean squared prediction error (RMSPE) and the mean absolute prediction error (MAPE), both of which should tend to zero. Prediction uncertainty accuracy is measured by the mean and standard deviation (SD) of the prediction z-score data (mean.ZS and SD.ZS, respectively). These z-scores are defined as:

$$\text{z-score}_{pred}(s) = (y(s) - \hat{y}(s)) / \sigma_{pred}(s)$$

where for unbiased prediction standard errors, the mean and SD of the z-scores should tend to zero and unity, respectively. As would be expected, GW regression provides the best set of results and there is near exact correspondence between the two global regression results. Finally, a map depicting the GW regression predictions is given in Figure 14b. Commands to conduct this model performance analysis, just for the GW regression model, are as follows:

```
R> RMSPE.gwr <-
+   (mean((ewhp_valid.spdf$PurPrice - gwr.pred$SDF$prediction)^2))^0.5
R> MAPE.gwr <- mean(abs(ewhp_valid.spdf$PurPrice - gwr.pred$SDF$prediction))
R> zscore.gwr <- (ewhp_valid.spdf$PurPrice - gwr.pred$SDF$prediction)/
+   (gwr.pred$SDF$prediction_var)^0.5
R> MeanZ.gwr <- mean(zscore.gwr)
R> SDZ.gwr <- (var(zscore.gwr))^0.5
R> gwr.pred$SDF$prediction <- gwr.pred$SDF$prediction/1000
R> quick.map(gwr.pred$SDF, "prediction", "1000's UK Pounds",
+   "GW Regression House Price predictions")
```

9. Comparisons with spgwr, gwrr and McSpatial

In this section, we compare the functions of **GWmodel** with those found in the related packages of **spgwr** (Version 0.6-24), **gwrr** (Version 0.2-1) and **McSpatial** (Version 2.0). Here we establish where they duplicate (or have strong similarities with) each other and where they diverge.

	GWmodel	spgwr	gwrr	McSpatial
Kernel functions:	Box-car, Bi-square, Tri-cube, Gaussian and Exponential	Bi-square, Tri-cube and Gaussian	Gaussian and Exponential	Rectangular, Triangular, Epanechnikov, Bi-square, Tri-cube, Tri-weight and Gaussian
Adaptive bandwidth?	Yes	Yes with <code>gwr.adapt</code>	No	No
Fixed bandwidth?	Yes	Yes	Yes	Yes
Spatial distance metrics:	Euclidean, Great Circle and Minkowski	Euclidean and Great Circle	Euclidean	Euclidean and Great Circle
Functions for weights matrix computation?	Yes with <code>gw.dist</code>	Yes with <code>gwr.bisquare</code> , <code>gwr.tricube</code> and <code>gwr.gauss</code>	No	Yes with <code>makew</code>

Table 3: Weighting matrix specifications.

	GWmodel	spgwr	gwrr	McSpatial
Basic? Statistics:	Yes with <code>gwss</code> Mean, std. deviation, variance, skew, coeff. of variation, covariance, Pearsons's correlation	Yes with <code>gw.cov</code> Mean, std. deviation, std. error of mean, std. difference of global mean, std. difference of local mean, covariance, Pearson's correlation	No	No
Robust? statistics:	Yes with <code>gwss</code> Median, inter-quartile range, quantile imbalance, Spearman's rank correlation	No	No	No
Monte-Carlo tests?	Yes with <code>montecarlo.gwss</code>	No	No	No

Table 4: GW summary statistics functions.

This comparison is given as a series of tables. Weighting matrix specifications are summarized in Table 3. Functions for GW summary statistics are summarized in Table 4. Functions for basic GW regression are summarized in Table 5. Functions for generalized GW regression are summarized in Table 6. Functions for other GW regressions are given in Table 7. Functions for addressing collinearity issues in GW regressions are given in Table 8. Lastly, functions for other GW models (including a GW PCA and a GW discriminant analysis) are given in Table 9. Clearly, **GWmodel** provides a more extensive set of GW models and associated tools than any existing package.

	GWmodel	spgwr	gwrr	McSpatial
Basic GW regression?	Yes with gwr.basic	Yes with gwr	Yes with gwr.est	Yes with cparlwr
CV bandwidth?	Yes with bw.gwr	Yes with gwr.sel	Yes with gwr.bw.est	Yes with cparlwrgrid
Generalized CV bandwidth?	No	No	No	Yes with cparlwrgrid
AICc bandwidth?	Yes with bw.gwr	Yes with gwr.sel	No	No
Investigation of CV bandwidth?	Yes with gwr.cv and gwr.cv.contrib	No	No	No
Model selection tools?	Yes with model.selection.gwr , model.sort.gwr and model.view.gwr	No	No	No
Local vs. global regression tests?	Yes with gwr.basic	Yes with LMZ.F3GWR.test	No	No
Monte-Carlo test?	Yes with montecarlo.gwr	No	No	No
Multiple hypothesis tests?	Yes with gwr.t.adjust	No	No	No
Moran's I test?	No	Yes with gwr.morantest	No	No
Prediction variances (or std. errors)?	Yes with gwr.predict	Yes with gwr	No	No
Parallel computing options?	No	Yes with gwr	No	No

Table 5: Basic GW regression functions.

	GWmodel	spgwr	gwrr	McSpatial
Generalized GW regression?	Yes with gwr.generalised	Yes with ggwr	No	Yes with cparlogit , cparmlogit and cparprobit
Family/link:	Binomial, Poisson	As for glm	No	Logit, multinomial, probit
CV bandwidth?	Yes with bw.ggwr	Yes with ggwr.sel	No	Yes with cparlwrgrid
Generalized CV bandwidth?	No	No	No	Yes with cparlwrgrid
AICc bandwidth?	Yes with bw.ggwr	No	No	No
Investigative tools for CV bandwidth?	Yes with ggwr.cv and ggwr.cv.contrib	No	No	No

Table 6: Generalized GW regression functions.

	GWmodel	spgwr	gwrr	McSpatial
Robust GW regression?	Yes with <code>gwr.robust</code>	No	No	No
Heteroskedastic GW regression?	Yes with <code>gwr.hetero</code>	No	No	No
Mixed (semipara.) GW regression?	Yes with <code>gwr.mixed</code>	No	No	Yes with <code>semip</code>
Quantile GW regression?	No	No	No	Yes with <code>qregcpar</code>

Table 7: Other GW regression functions.

	GWmodel	spgwr	gwrr	McSpatial
Local correlations?	Yes with <code>gwss</code> or <code>gwr.collin.diagno</code>	Yes with <code>gw.cov</code>	No	No
Local VIFs?	Yes with <code>gwr.collin.diagno</code>	No	No	No
Local VDPs?	Yes with <code>gwr.collin.diagno</code>	No	Yes with <code>gwr.vdp</code>	No
Local condition numbers?	Yes with <code>gwr.lcr</code> or <code>gwr.collin.diagno</code>	No	Yes with <code>gwr.vdp</code>	No
GW regression with a global ridge term (user-specified)?	Yes with <code>gwr.lcr</code> , <code>bw.gwr.lcr</code> , <code>gwr.lcr.cv</code> , <code>gwr.lcr.cv.contrib</code>	No	Yes with <code>gwrr.est</code>	No
GW regression with a global ridge term (estimated)?	No	No	Yes with <code>gwrr.est</code>	No
GW regression with local ridge terms?	Yes with <code>gwr.lcr</code> , <code>bw.gwr.lcr</code> , <code>gwr.lcr.cv</code> , <code>gwr.lcr.cv.contrib</code>	No	No	No
GW lasso?	No	No	Yes with <code>gw1.est</code>	No

Table 8: Functions for addressing collinearity issues in GW regression.

	GWmodel	spgwr	gwrr	McSpatial
GW discriminant analysis?	Yes with <code>bw.gwda</code> and <code>gwda</code>	No	No	No
GW PCA?	Yes with <code>bw.gwpca</code> , <code>gwpca</code> , <code>gwpca.cv</code> , <code>gwpca.cv.contrib</code> , <code>glyph.plot</code> , <code>check.components</code> , <code>plot.mcsims</code> , <code>montecarlo.gwpca.1</code> , <code>montecarlo.gwpca.2</code>	No	No	No
GW parallel coordinate plot?	Yes with <code>gw.pcplot</code>	No	No	No

Table 9: Other GW model functions.

10. Discussion

In this presentation of the **GWmodel** R package, we have demonstrated the use of various GW models to investigate and model different aspects of spatial heterogeneity. We have focused our presentation on basic and robust forms of GW summary statistics, GW PCA and GW regression. We have also provided important extensions to GW regression with respect to local collinearity issues and with respect to spatial prediction. However, we have not fully described all that is available in **GWmodel**. Key omissions include:

1. Functions to investigate bandwidth selection, where a cross-validation score can be found for a given bandwidth (`ggwr.cv`, `gwr.cv`, `gwr.lcr.cv`, `gwpca.cv`); and associated functions to find which observations contribute the most (and are potentially outlying) to this cross-validation score (`ggwr.cv.contrib`, `gwr.cv.contrib`, `gwr.lcr.cv.contrib`, `gwpca.cv.contrib`).
2. Functions implementing Monte Carlo tests for GW summary statistics (`montecarlo.gwss`), GW regression (`montecarlo.gwr`) and GW PCA (`montecarlo.gwpca.1`, `montecarlo.gwpca.2`). These functions test whether aspects of the GW model are significantly different to those that could be found under the global model, as artefacts of random variation in the data.
3. Functions for fitting generalized GW regression models (`gwr.generalised`, `ggwr.cv`, `ggwr.cv.contrib`, `bw.ggwr`).
4. Further functions for visualising GW PCA outputs (`glyph.plot`, `check.components`) and for locally visualising the original multivariate data (`gw.pcplot`).
5. A function for multiple hypothesis tests with GW regression (`gwr.t.adjust`).
6. Functions for fitting mixed (`gwr.mixed`) and heteroskedastic (`gwr.hetero`) GW regression.
7. Functions for conducting a GW discriminant analysis (`gwda`, `bw.gwda`).

The use of these functions will be presented elsewhere. Future iterations of **GWmodel** will include many more functions, such as those needed to implement GW boxplots and GW variograms. The GW modelling paradigm itself continues to evolve, with not only new GW models, such as the robust GW regressions of [Zhang and Mei \(2011\)](#) or the GW quantile regression of [Chen, Deng, Yang, and Matthews \(2012\)](#); but also, novel usages of existing GW models, such as using GW PCA to detect multivariate spatial outliers ([Harris, Brunson, Charlton, Juggins, and Clarke 2014a](#)) or using GW PCA to direct sample re-design ([Harris, Clarke, Juggins, Brunson, and Charlton 2014b](#)).

Acknowledgments

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

References

- Akaike H (1973). “Information Theory and an Extension of the Maximum Likelihood Principle.” In BN Petrov, F Csaki (eds.), *2nd Symposium on Information Theory*, pp. 267–281. Akademiai Kiado, Budapest.
- Assuncao R (2003). “Space Varying Coefficient Models for Small Area Data.” *Environmetrics*, **14**, 453–473.
- Belsley DA, Kuh E, Welsch RE (1980). *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*. John Wiley & Sons.
- BioMedware (2011). *SpaceStat User Manual, Version 2.2*. New York. URL <http://www.biomedware.com/>.
- Bitter C, Mulligan G, Dall’Erba S (2007). “Incorporating Spatial Variation in Housing Attribute Prices: A Comparison of Geographically Weighted Regression and the Spatial Expansion Method.” *Journal of Geographical Systems*, **9**(1), 7–27.
- Bivand R, Yu D, Nakaya T, Garcia-Lopez MA (2013). *spgwr: Functions for Computing Geographically Weighted Regressions*. R package version 0.6-22, URL <http://CRAN.R-project.org/package=spgwr>.
- Bowman A (1984). “An Alternative Method of Cross-Validation for the Smoothing of Density Estimates.” *Biometrika*, **71**, 353–360.
- Brunsdon C, Charlton M, Harris P (2012). “Living with Collinearity in Local Regression Models.” In *Proceedings of the 10th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*. Brasil.
- Brunsdon C, Fotheringham AS, Charlton M (1996). “Geographically Weighted Regression: A Method for Exploring Spatial Nonstationarity.” *Geographical Analysis*, **28**, 281–289.
- Brunsdon C, Fotheringham AS, Charlton M (1998). “Geographically Weighted Regression – Modelling Spatial Non-Stationarity.” *Journal of the Royal Statistical Society D*, **47**(3), 431–443.
- Brunsdon C, Fotheringham AS, Charlton M (1999). “Some Notes on Parametric Significance Tests for Geographically Weighted Regression.” *Journal of Regional Science*, **39**(3), 497–524.
- Brunsdon C, Fotheringham AS, Charlton M (2002). “Geographically Weighted Summary Statistics – A Framework for Localised Exploratory Data Analysis.” *Computers, Environment and Urban Systems*, **26**(6), 501–524.
- Brunsdon C, Fotheringham AS, Charlton M (2007). “Geographically Weighted Discriminant Analysis.” *Geographical Analysis*, **39**(4), 376–396.
- Charlton M, Fotheringham AS, Brunsdon C (2003). *GWR 3: Software for Geographically Weighted Regression*. NCG, National University of Ireland Maynooth.

- Chen VYJ, Deng WS, Yang TC, Matthews SA (2012). “Geographically Weighted Quantile Regression (GWQR): An Application to U.S. Mortality Data.” *Geographical Analysis*, **44**(2), 134–150.
- Cleveland WS (1979). “Robust Locally Weighted Regression and Smoothing Scatterplots.” *Journal of the American Statistical Association*, **74**, 823–836.
- Cleveland WS, Devlin SJ (1988). “Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting.” *Journal of the American Statistical Association*, **83**(403), 596–610.
- Cressie N (1989). “The Many Faces of Spatial Prediction.” In M Armstrong (ed.), *Geostatistics*, volume 1, pp. 163–176. Dordrecht, Kluwer.
- Dykes J, Brunsdon C (2007). “Geographically Weighted Visualization: Interactive Graphics for Scale-Varying Exploratory Analysis.” *IEEE Transactions on Visualization and Computer Graphics*, **13**(6), 1161–1168.
- ESRI (2013). *ArcGIS Resource Center, Version 10.2*. Environmental Systems Resource Institute, Redlands, California. URL <http://www.arcgis.com/>.
- Finley AO (2011). “Comparing Spatially-Varying Coefficient Models for Analysis of Ecological Sata with Non-Stationary and Anisotropic Residual Dependence.” *Ecology and Evolution*, **2**, 143–154.
- Foley P, Demsar U (2013). “Using Geovisual Analytics to Compare the Performance of Geographically Weighted Discriminant Analysis versus its Global Counterpart, Linear Discriminant Analysis.” *International Journal of Geographical Information Systems*, **27**(4), 633–661.
- Fotheringham AS, Brunsdon C, Charlton M (2002). *Geographically Weighted Regression: The Analysis of Spatially Varying Relationships*. John Wiley & Sons, Chichester.
- Fox J, Monette G (1992). “Generalized Collinearity Diagnostics.” *Journal of the American Statistical Association*, **87**(417), 178–183.
- Fox J, Weisberg S (2011). *An R Companion to Applied Regression*. 2nd edition. Sage, Thousand Oaks.
- Frank IE, Friedman JH (1993). “A Statistical View of Some Chemometrics Regression Tools.” *Technometrics*, **35**(2), 109–135.
- Freeman E, Frescino T (2009). *ModelMap: Modeling and Map Production Using Random Forest and Stochastic Gradient Boosting*. USDA Forest Service, Rocky Mountain Research Station, Ogden, UT, United States of America.
- Gao X, Asami Y, Chung CJF (2006). “An Empirical Evaluation of Spatial Regression Models.” *Computers & Geosciences*, **32**(8), 1040–1051.
- Gelfand A, Kim H, Sirmans C, Banerjee S (2003). “Spatial Modelling with Spatially Varying Coefficient Processes.” *Journal of American Statistical Association*, **98**, 387–396.

- Griffith DA (2008). “Spatial-Filtering-Based Contributions to a Critique of Geographically Weighted Regression (GWR).” *Environment and Planning A*, **40**(11), 2751–2769.
- Harris P, Brunsdon C (2010). “Exploring Spatial Variation and Spatial Relationships in a Freshwater Acidification Critical Load Data Set for Great Britain Using Geographically Weighted Summary Statistics.” *Computers & Geosciences*, **36**(1), 54–70.
- Harris P, Brunsdon C, Charlton M (2011a). “Geographically Weighted Principal Components Analysis.” *International Journal of Geographical Information Science*, **25**(10), 1717–1736.
- Harris P, Brunsdon C, Charlton M, Juggins S, Clarke A (2014a). “Multivariate Spatial Outlier Detection Using Robust Geographically Weighted Methods.” *Mathematical Geosciences*, **46**(1), 1–31.
- Harris P, Brunsdon C, Fotheringham AS (2011b). “Links, Comparisons and Extensions of the Geographically Weighted Regression Model When Used as a Spatial Predictor.” *Stochastic Environmental Research and Risk Assessment*, **25**(2), 123–138.
- Harris P, Charlton M, Fotheringham AS (2010a). “Moving Window Kriging with Geographically Weighted Variograms.” *Stochastic Environmental Research and Risk Assessment*, **24**(8), 1193–1209.
- Harris P, Clarke A, Juggins S, Brunsdon C, Charlton M (2014b). “Geographically Weighted Methods and Their Use in Network Re-Designs for Environmental Monitoring.” *Stochastic Environmental Research and Risk Assessment*, **28**(7), 1869–1887.
- Harris P, Fotheringham AS, Crespo R, Charlton M (2010b). “The Use of Geographically Weighted Regression for Spatial Prediction: An Evaluation of Models Using Simulated Data Sets.” *Mathematical Geosciences*, **42**(6), 657–680.
- Harris P, Juggins S (2011). “Estimating Freshwater Acidification Critical Load Exceedance Data for Great Britain Using Space-Varying Relationship Models.” *Mathematical Geosciences*, **43**(3), 265–292.
- Hoerl AE (1962). “Application of Ridge Analysis to Regression Problems.” *Chemical Engineering Progress*, **58**(3), 54–59.
- Hoerl AE, Kennard RW (1970). “Ridge Regression: Biased Estimation for Nonorthogonal Problems.” *Technometrics*, **12**(1), 55–67.
- Hurvich CM, Simonoff JS, Tsai CL (1998). “Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion.” *Journal of the Royal Statistical Society B*, **60**(2), 271–293.
- Jolliffe IT (2002). *Principal Component Analysis*. 2nd edition. Springer-Verlag, New York.
- Kavanagh A, Fotheringham S, Charlton M (2006). “A Geographically Weighted Regression Analysis of the Election Specific Turnout Behaviour in the Republic of Ireland.” In *Elections, Public Opinion and Parties Conference, Nottingham 8th to 10th September 2006*.
- Kestens Y, Thériault M, Rosiers FD (2006). “Heterogeneity in Hedonic Modelling of House Prices: Looking at Buyers’ Household Profiles.” *Journal of Geographical Systems*, **8**(1), 61–96.

- Lee TS (1987). “Algorithm AS 223: Optimum Ridge Parameter Selection.” *Journal of the Royal Statistical Society C*, **36**(1), 112–118.
- Leung Y, Mei CL, Zhang WX (2000). “Statistical Tests for Spatial Nonstationarity Based on Geographically Weighted Regression.” *Environment and Planning A*, **32**, 9–32.
- Lloyd CD (2010a). “Analysing Population Characteristics Using Geographically Weighted Principal Components Analysis: A Case Study of Northern Ireland in 2001.” *Computers, Environment and Urban Systems*, **34**, 389–399.
- Lloyd CD (2010b). “Nonstationary Models for Exploring and Mapping Monthly Precipitation in the United Kingdom.” *International Journal of Climatology*, **30**(3), 390–405.
- Lloyd CD, Shuttleworth I (2005). “Analysing Commuting Using Local Regression Techniques: Scale, Sensitivity, and Geographical Patterning.” *Environment and Planning A*, **37**, 81–103.
- McMillen D (2013). **McSpatial: Nonparametric Spatial Data Analysis**. R package version 2.0, URL <http://CRAN.R-project.org/package=McSpatial>.
- Nakaya T, Fotheringham AS, Brunsdon C, Charlton M (2005). “Geographically Weighted Poisson Regression for Disease Association Mapping.” *Statistics in Medicine*, **24**(17), 2695–2717.
- Neuwirth E (2011). **RColorBrewer: ColorBrewer Palettes**. R package version 1.0-5, URL <http://CRAN.R-project.org/package=RColorBrewer>.
- O’Brien RM (2007). “A Caution Regarding Rules of Thumb for Variance Inflation Factors.” *Quality & Quantity*, **41**(5), 673–690.
- Openshaw S, Charlton M, Wymer C, Craft A (1987). “A Mark 1 Geographical Analysis Machine for the Automated Analysis of Point Data Sets.” *International Journal of Geographical Information System*, **1**(4), 335–358.
- Páez A, Farber S, Wheeler D (2011). “A Simulation-Based Study of Geographically Weighted Regression as a Method for Investigating Spatially Varying Relationships.” *Environment and Planning-Part A*, **43**(12), 2992–3010.
- Páez A, Long F, Farber S (2008). “Moving Window Approaches for Hedonic Price Estimation: An Empirical Comparison of Modelling Techniques.” *Urban Studies*, **45**(8), 1565–1581.
- Pebesma EJ (2004). “Multivariable Geostatistics in S: The **gstat** Package.” *Computers & Geosciences*, **30**(7), 683–691.
- Rangel TF, Diniz-Filho JAF, Bini LM (2010). “**SAM: A Comprehensive Application for Spatial Analysis in Macroecology**.” *Ecography*, **33**(1), 46–50.
- R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Robinson A (2013). *Geovisualization of the 2004 Presidential Election*. URL <http://www.personal.psu.edu/users/a/c/acr181/election.html>.

- Rousseeuw PJ (1985). “Multivariate Estimation with High Breakdown Point.” *Mathematical Statistics and Applications*, **B**, 283–297.
- Tibshirani R (1996). “Regression Shrinkage and Selection via the Lasso.” *Journal of the Royal Statistical Society B*, **58**, 267–288.
- Varmuza K, Filzmoser P (2009). *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press.
- Waller LA, Zhu L, Gotway CA, Gorman DM, Grunewald PJ (2007). “Quantifying Geographic Variations in Associations Between Alcohol Distribution and Violence: A Comparison of Geographically Weighted Regression and Spatially Varying Coefficient Models.” *Stochastic Environmental Research and Risk Assessment*, **21**, 573–588.
- Wheeler D (2007). “Diagnostic Tools and a Remedial Method for Collinearity in Geographically Weighted Regression.” *Environment and Planning A*, **39**(10), 2464–2481.
- Wheeler D (2009). “Simultaneous Coefficient Penalization and Model Selection in Geographically Weighted Regression: the Geographically Weighted Lasso.” *Environment and Planning A*, **41**(3), 722–742.
- Wheeler D (2010). “Visualizing and Diagnosing Coefficients from Geographically Weighted Regression.” In B Jiang, X Yao (eds.), *Geospatial Analysis and Modelling of Urban Structure and Dynamics*. Springer-Verlag.
- Wheeler D (2013a). “Geographically Weighted Regression.” In M Fischer, P Nijkamp (eds.), *Handbook of Regional Science*. Springer-Verlag.
- Wheeler D (2013b). *gwrr: Fits Geographically Weighted Regression Models with Diagnostic Tools*. R package version 0.2-1, URL <http://CRAN.R-project.org/package=gwrr>.
- Wheeler D, Calder C (2007). “An Assessment of Coefficient Accuracy in Linear Regression Models with Spatially Varying Coefficients.” *Journal of Geographical Systems*, **9**, 145–166.
- Wheeler D, Tiefelsdorf M (2005). “Multicollinearity and Correlation among Regression Coefficients in Geographically Weighted Regression.” *Journal of Geographical Systems*, **7**(2), 161–187.
- Wheeler D, Waller LA (2009). “Comparing Spatially Varying Coefficient Models: A Case Study Examining Violent Crime Rates and Their Relationships to Alcohol Outlets and Illegal Drug Arrests.” *Journal of Geographical Systems*, **11**, 1–22.
- Zhang H, Mei C (2011). “Local Least Absolute Deviation Estimation of Spatially Varying Coefficient Models: Robust Geographically Weighted Regression Approaches.” *International Journal of Geographical Information Science*, **25**(9), 1467–1489.

Affiliation:

Isabella Gollini

Department of Civil Engineering

University of Bristol

Queen's Building

Bristol BS8 1TR, United Kingdom

E-mail: isabella.gollini@bristol.ac.uk

URL: <http://www.bristol.ac.uk/engineering/people/isabella-gollini/>